L02: Scaling from one computer to thousands.

ANLY 502: Massive Data Fundamental

Simson Garfinkel & Ghaleb Abdulla

February 8, 2016

Outline for today's lesson

Student Presentations:

- Yu YU — Phonetic analytics technology and big data: real-world cases
- Jiayao Wang — Schema.org: Evolution of Structured Data on the Web
- John Hotchkiss — Accountability in algorithmic decision making

FIVE MINUTES EACH!

Scaling beyond a single computer

- The Cloud
- Hadoop Architecture — Hadoop, Yarn, and HDFS
- Amazon EC2 & Elastic Map Reduce

Institutional Review Boards

Previewing PS03

https://en.wikipedia.org/wiki/List_of_cloud_types#/media/File:Cirrus_clouds2.jpg

The Cloud…

The Cloud

Typical machine room

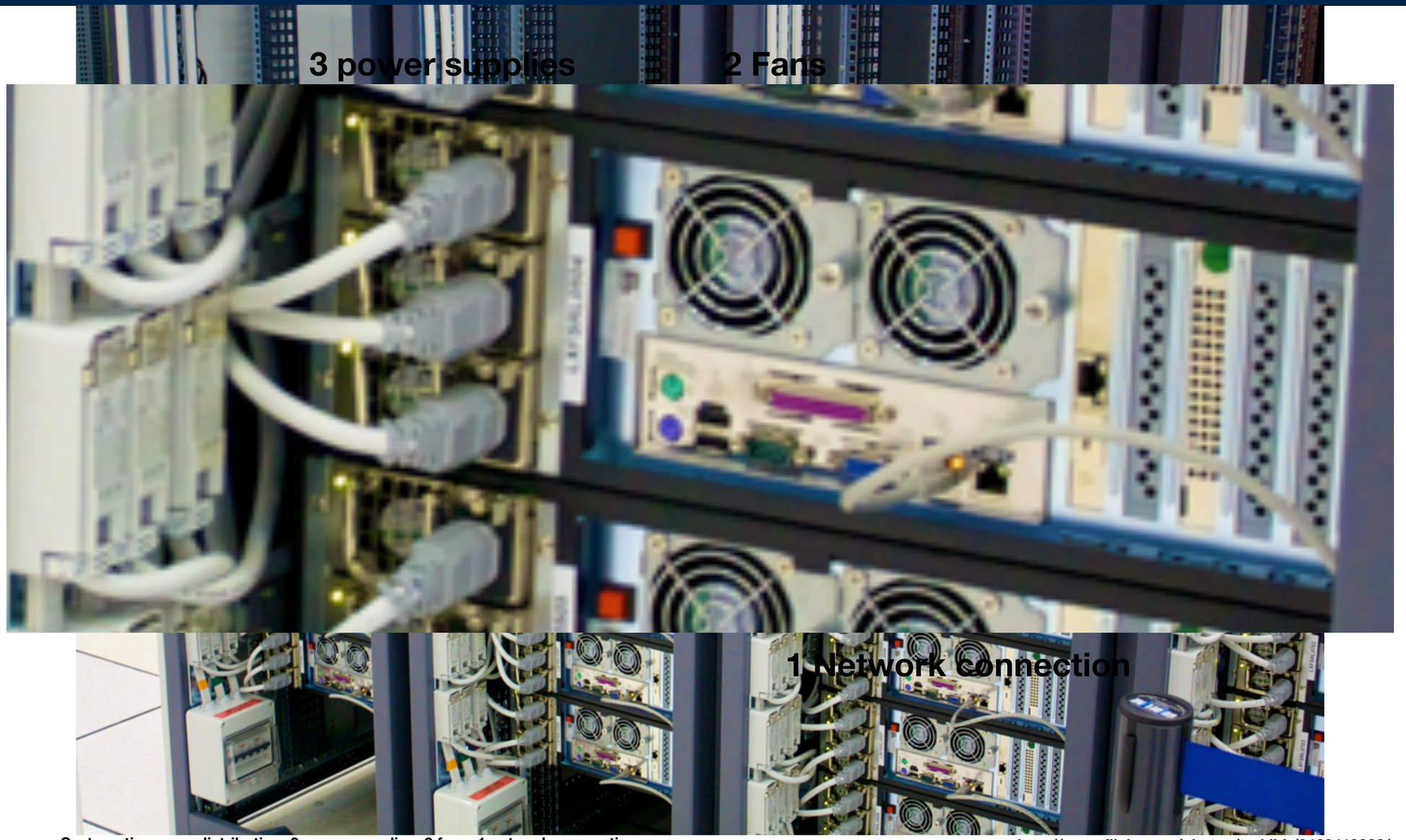
http://www.flickr.com/photos/torkildr/3462606643/sizes/l/in/photostream/

The Cloud

Many identical servers — each with disk, CPU and network

http://www.flickr.com/photos/torkildr/3462607995/in/photostream/

The Cloud



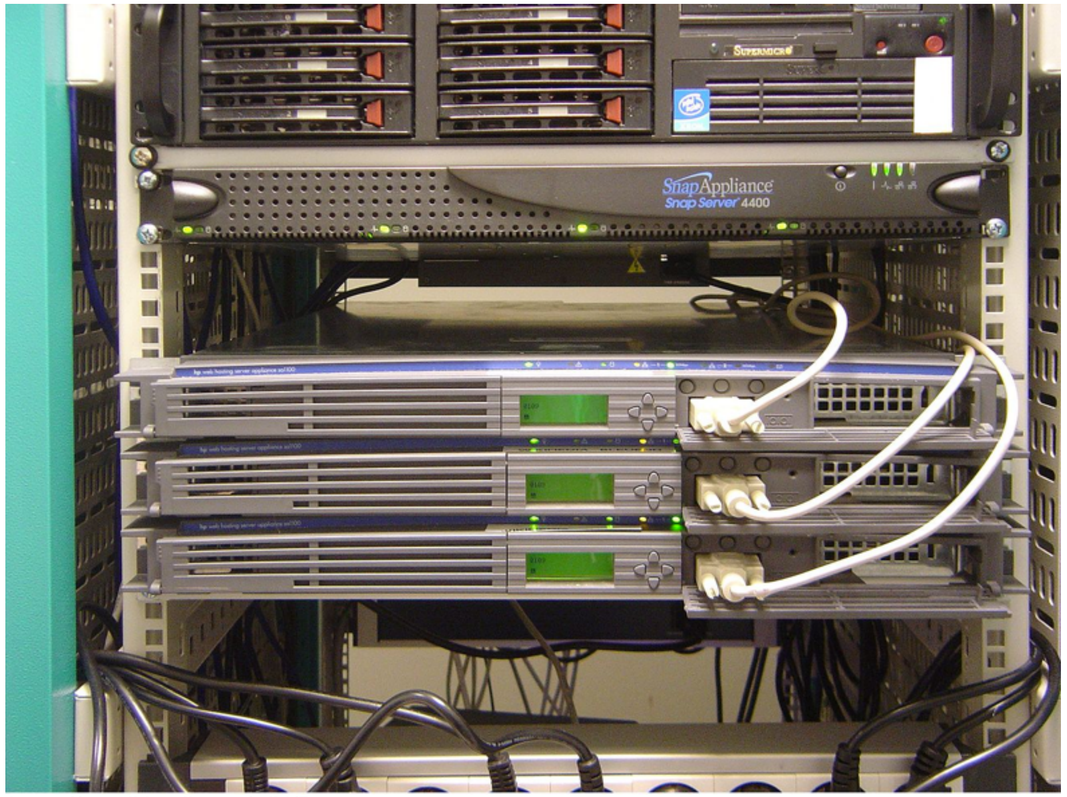
Systematic power distribution; 3 power supplies; 2 fans; 1 network connection

http://www.flickr.com/photos/torkildr/3463419826/

Early machine rooms: an equipment menagerie

This created significant manageability problems:

- Each machine had a distinct hardware and software configuration
- When a machine failed, it's services couldn't be readily moved to another.



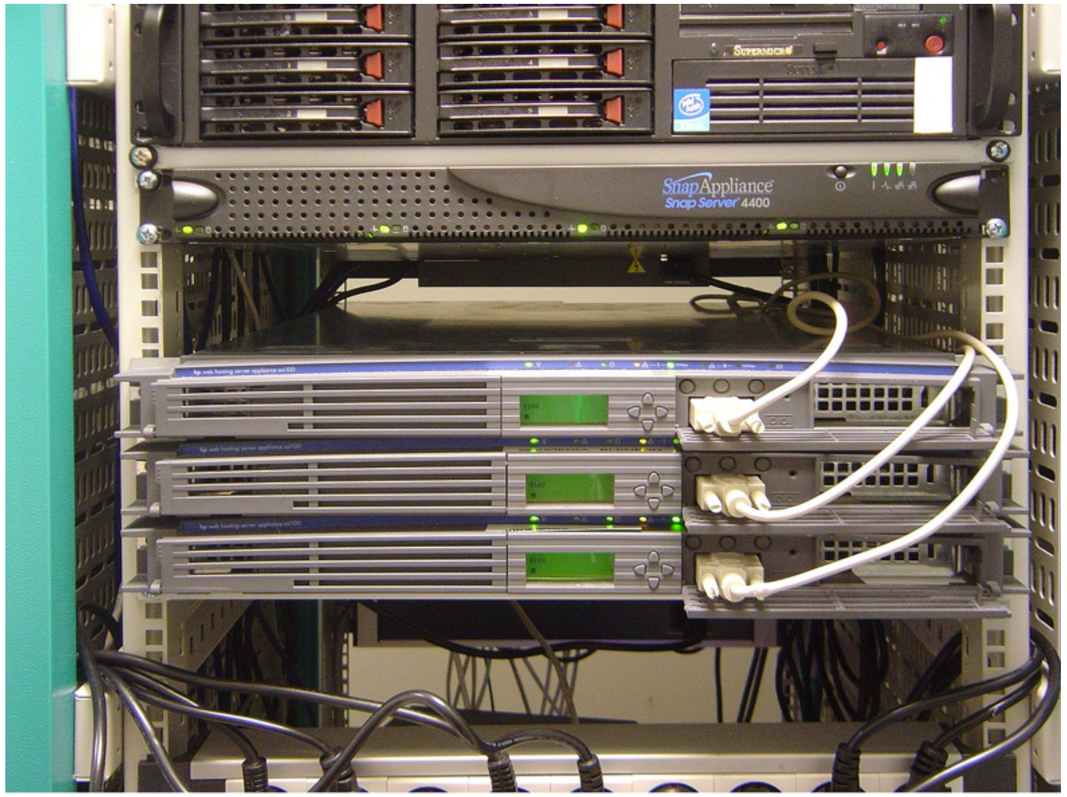
https://en.wikipedia.org/wiki/Data_center

1990s

Modern machine rooms have racks and racks of identical equipment.

Advantages:

- Consistent wiring plan.
- No computer is unique; easy fail-over and replacement



https://en.wikipedia.org/wiki/Data_center

1990s

2005

Modern data centers virtualize servers, storage, and networks.

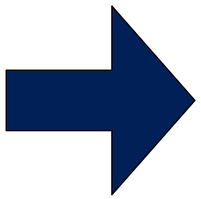
Server virtualization makes one server look like many

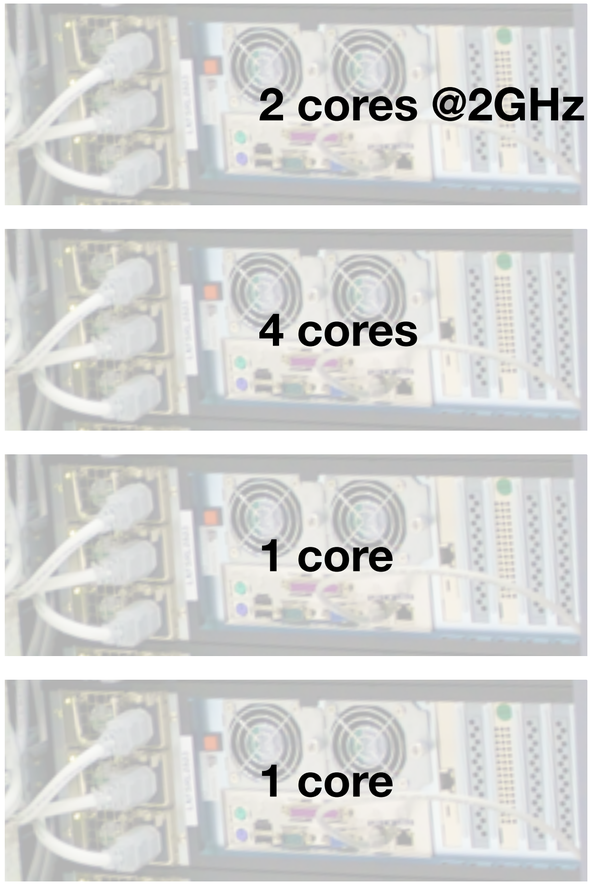
Advantages:

- Better hardware utilization (most servers do not run at 100%)
- Better scaling: if a server needs more CPU, give it more cores.

“Host” server
100% utilization

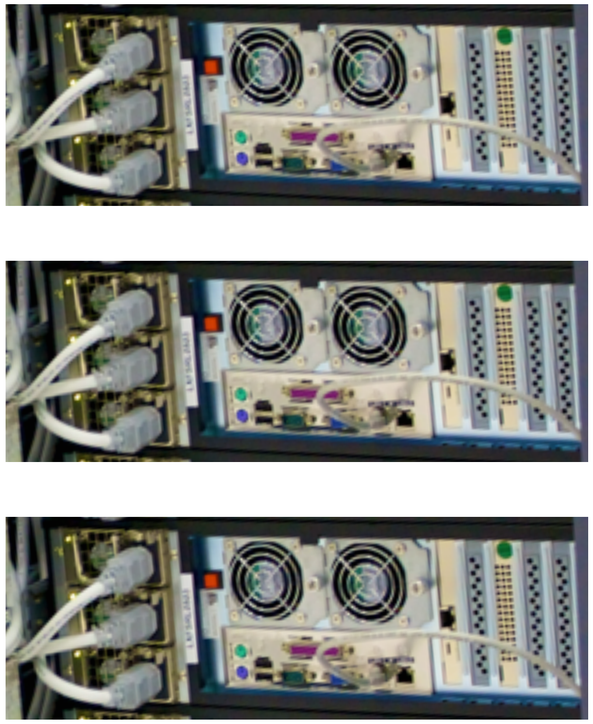
8 cores @ 2GHz





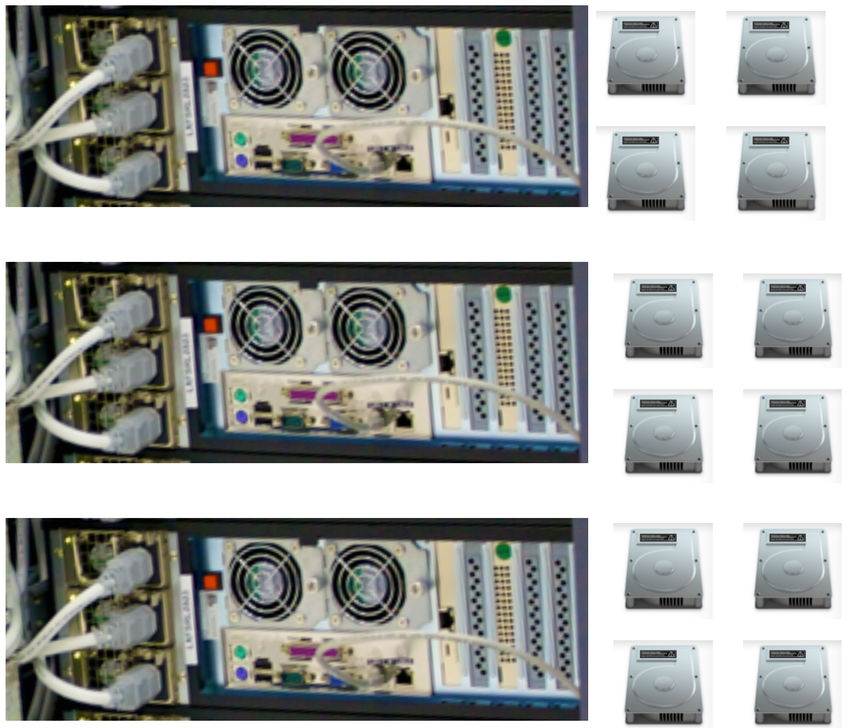
“Guest” operating systems
25% utilization each

This lets a few machines simulate many domains.



bikes.com	harry.pt	litter.org
lights.com	lead.pb	pete.sa
dinner.org	silver.ag	tanen.ak
nara.il	gold.au	simson.net
gizmo.as	carbon.cc	aneu.eu
john.au	orange.bo	nato.int
gilory.em	nov.em	jake.mn
tallen.ak	kenny.rf	jack.ac
when.ru	what.ru	web.mobi
where.org	kill.bill	sam.nom

Storage virtualization uses multiple servers to create the appearance of highly reliable storage arrays.

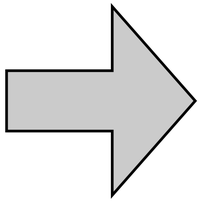


4 x 2TB

4 x 2TB

4 x 2TB

=24TB Raw



Reserve Storage

Advantages:

- Easy to allocate a new virtual drive (“volume”)
- Redundancy
- “Snapshot” and “Clone”
- Easy sharing of read-only volumes between servers.

Redundancy protects data against drive failure.

Every drive will eventually fail.

RAID — Redundant Array of Inexpensive Drives

- Stores data and “parity bits” across drives.
- Typical overhead: 16%-50% — 3-6 drives in a *RAID Set.*
  - —*RAID5 — Can tolerate the failure of 1 drive.*
  - —*RAID6 — Can tolerate the failure of 2 drives.*
- Requires “rebuild” when a drive is replaced.
  - —*Drives frequently fail during rebuild.*
- Hard to add more storage

Store: “DATA”

“DA” “TA” f(“DATA”)

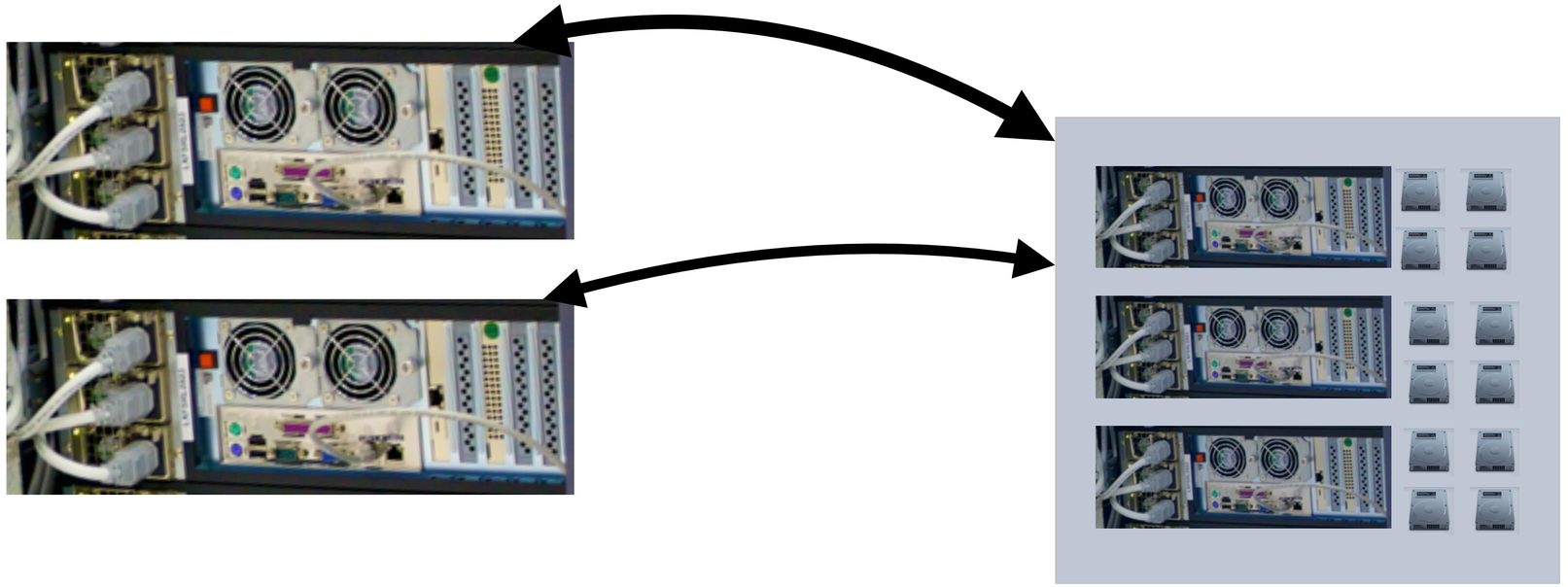
Replication

- Stores multiple copies of data on different drives.
- One copy gives protection against drive failures.
- Multiple copies gives increased performance
  - —*can read from multiple drives at once.*
- Typical overhead: 200% - 300% (or more)
- Requires object copying when a drive fails.
- Easy to add more storage

“DATA” “DATA” “DATA”

Virtualized storage appears as a remote “file server”

Virtual drives are accessed over the network.



Traditional file server protocols include:

- iSCSI — Block read/write protocol. (rw for single computer, ro for multiple)
- NFS — File read/write protocol. (rw or ro for multiple computers.)

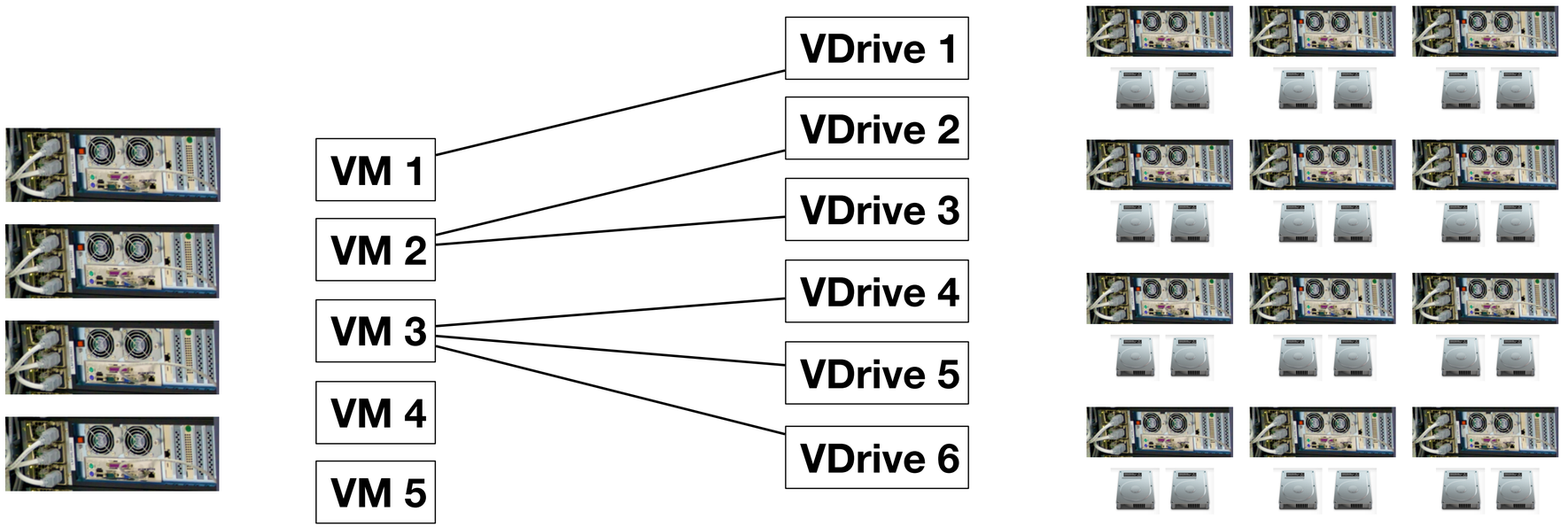
Big providers combine CPU and storage virtualization. Each virtual server runs on top of virtual storage.

Storage array holds:

- VM configurations
- VM drives.

If a drive fails, the array provides data availability

If a compute server fails, the VM restarts on another physical machine.



Virtualization allows for easy migration.

IBM Roadrunner @LANL - Completed in 2009

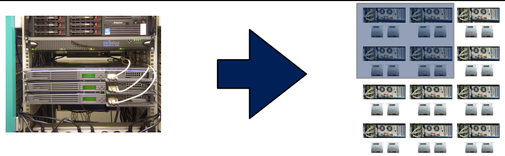
- World's fastest supercomputer
- $120 million
- 6912 Opterons
- 12,960 PowerXCell processors
- 130,464 cores
- 103.6 TB RAM
- 2PB storage
- 296 server cabinets
- full specs: http://ibm.co/1IdVgWC

Shut down March 31, 2013

- 444 megaflops per watt
  (c.f. 886 mflops per watt for others)
- Could be rebuilt in 2013 for $6 million; power cost $2.5 million/year
- Roadrunner was shredded.

http://bit.ly/1kE2lz7

There are two primary ways that the VMs and VDs are organized.



Separate virtualization of VMs and VDs:

- Lots of individual VMs.
  - —*Hosting provider catering to small and medium sized customers.*
  - —*Server consolidation*
- Scientific Computing:
  - —*A large job (e.g. simulation) is split into 10,000s of slices and run on each system.*
  - —*Sun Grid Engine (SGE) (Now the Open Source Grid Engine), Condor, pbs Torque*
  - —*High-performance Compute and I/O*
- Data-Centric Computing.
  - —*Combined virtualization of CPU and storage:*
  - —*Hadoop, MapReduce & Spark*

In practice, the distinction is not firm:

- Individual VMs may have code and data distributed to each system.
- MapReduce & Spark systems may have “compute only” nodes for extra processing.

Example of scientific computing: a typical supercomputer

Systems have:

- Separate CPU and storage
- Optimized for floating point.

Separate control and data:

- TCP/IP for control over 10gig
- Fibre channel SAN for disks
- Distributed file system, lets code quickly fetch data from high-performance disk servers.

Design goals:

- High component reliability = no failures.
- Predictable job execution.

Supercomputer "K computer" Takes First Place in World

Achieves world's best performance of 8.162 petaflops to lead TOP500 list

RIKEN,Fujitsu Limited

Tokyo, June 20, 2011

High performance CPUs

High performance storage

672 racks = 68,544 CPUS
8.162 petaflops

Example of a typical Hadoop rack

Systems have:

- CPU and storage in the same box

Integrated control and data:

- Data are distributed in different machines.
- Code goes to data to run.
- Results stored locally or sent to other nodes.

Design goals:

- Failure tolerant
  - —*If a computer fails once every 4 years, 1200 computers may average a failure every day!*
- Commodity hardware
  - —*Chapter to buy 4 computers with 16 cores thank 1 computer with 64 cores*

Commodity hardware

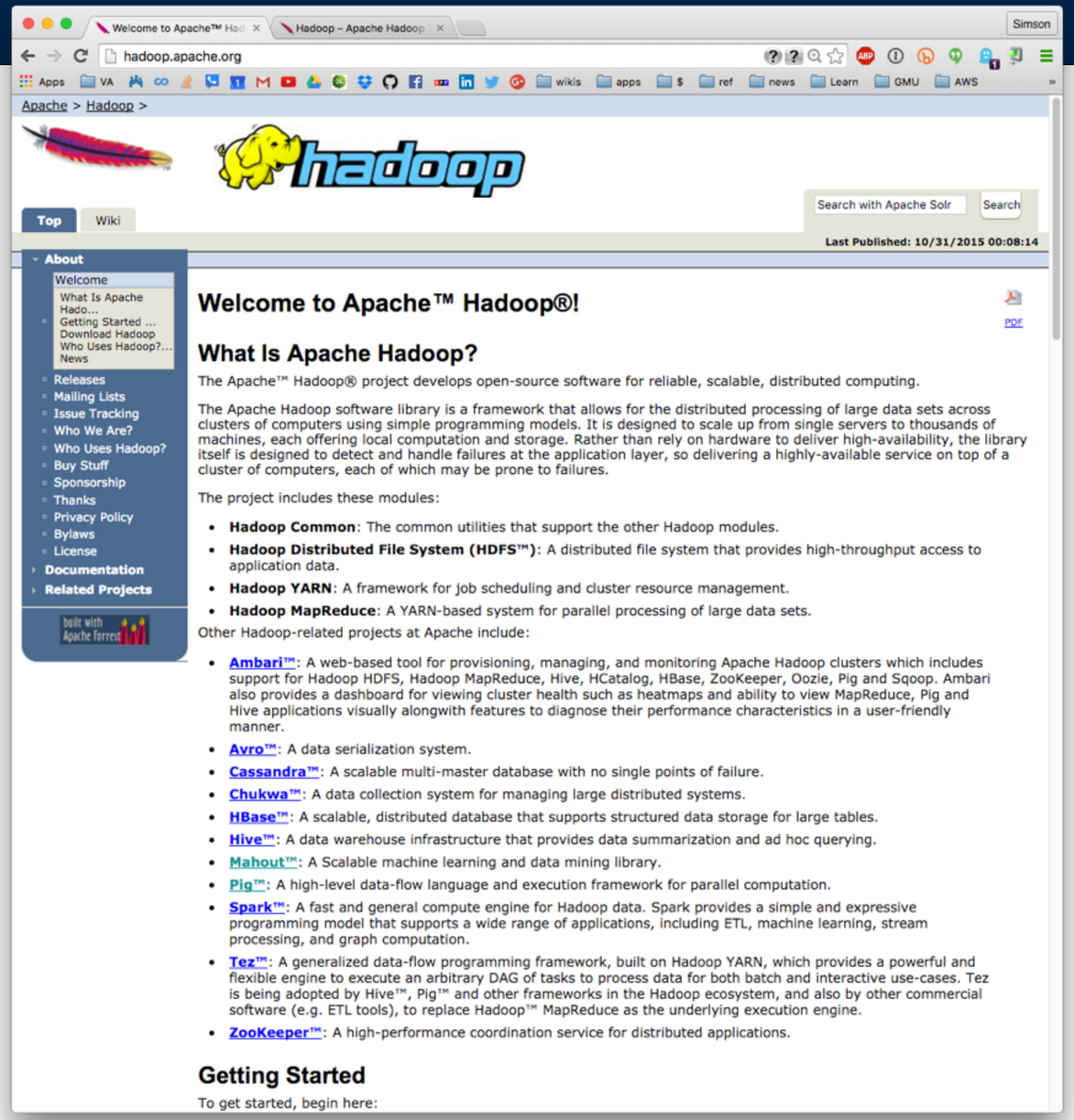
Hadoop Design and Architecture

Read the documentation!

The documentation is the definitive source of information about Hadoop.

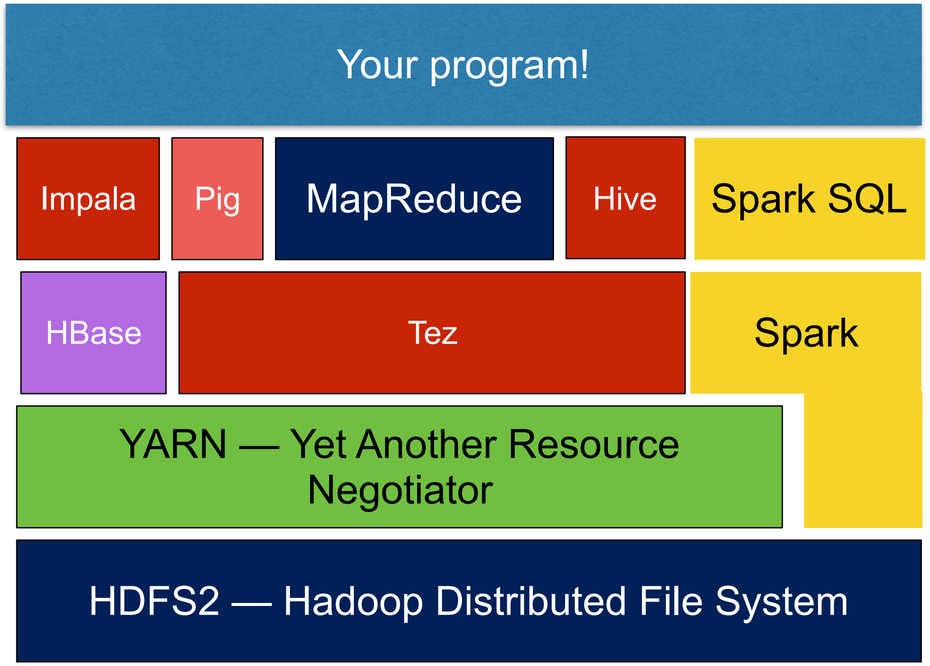
- Skip setting up.
- Concentrate on design and tutorial docs.
  - *—"Current" vs. "Stable"*
- References provide you

Be sure to read the docs for the version you are running.



Hadoop Architecture: We are using Hadoop Version 2

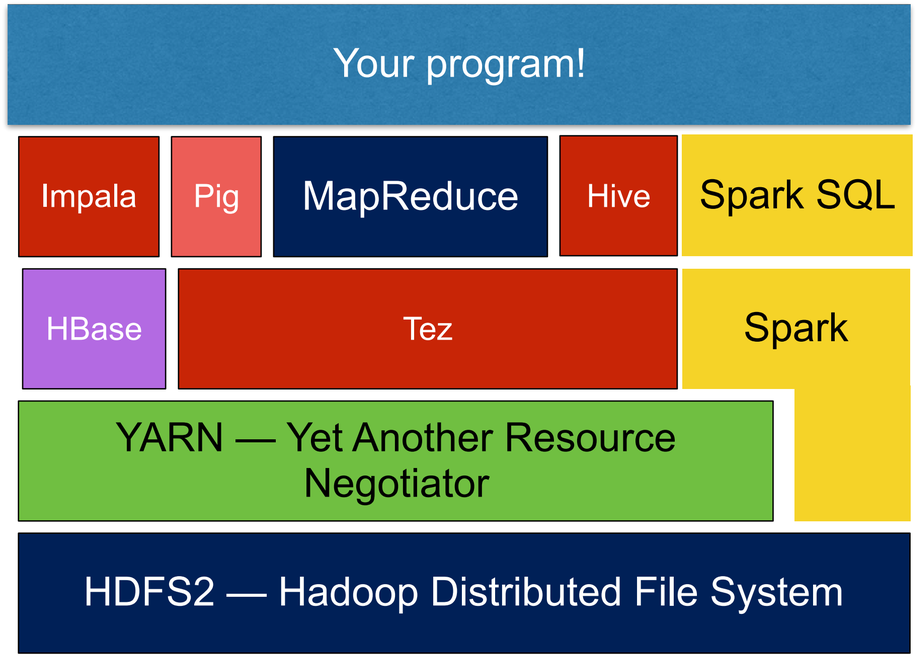
Hadoop is a collection of parts:



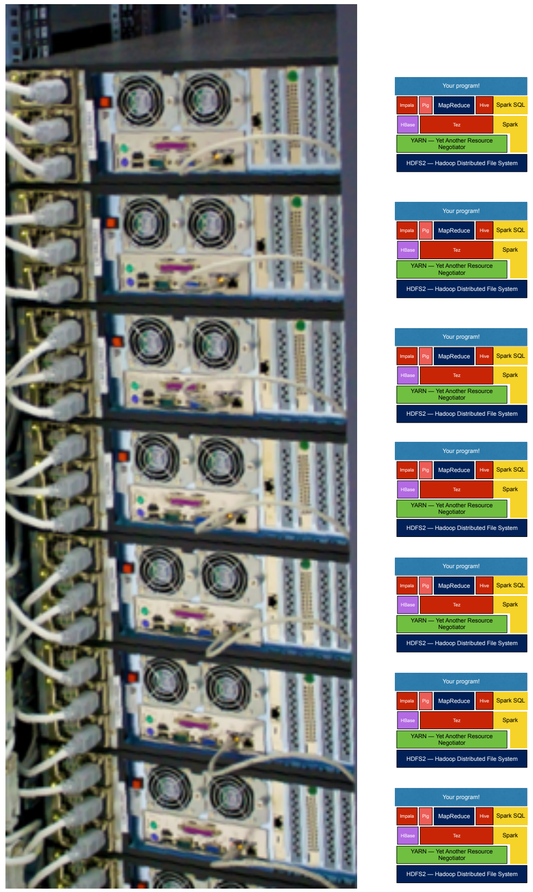
The stack runs on every node in the cluster

Each “node” is a rack-mounted computer:

- Running Linux
- Running multiple JVMs (1 per core)



Rack



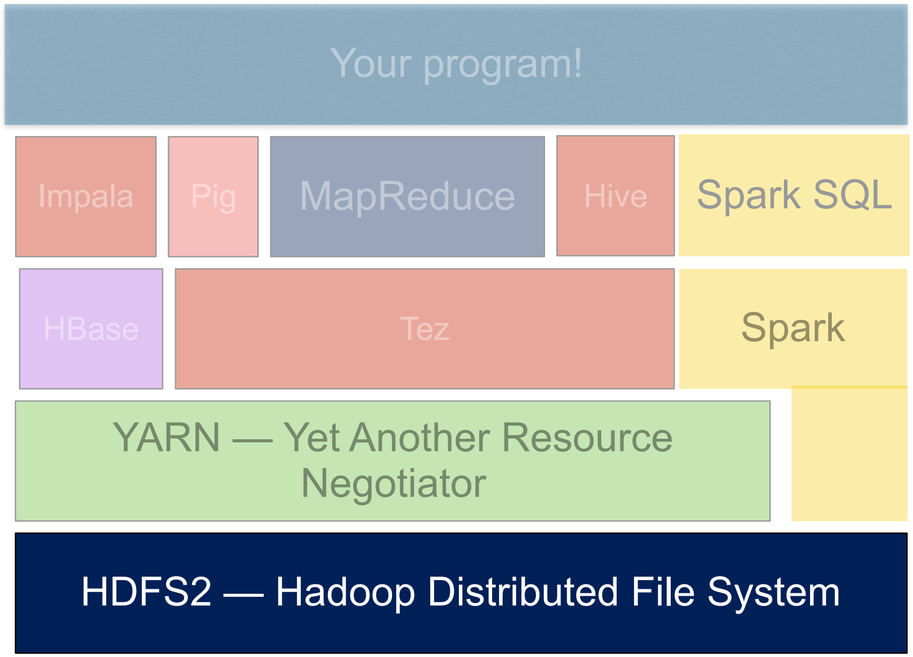
HDFS layer manages storage of data

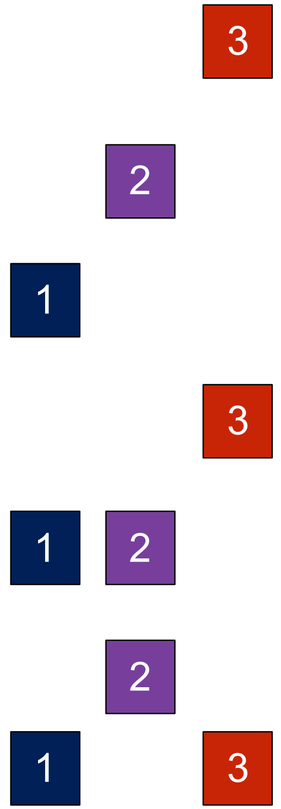
Data:

- split into blocks (64MB each)
- stored redundantly (no RAID)

Name node:

- Tracks location of every file

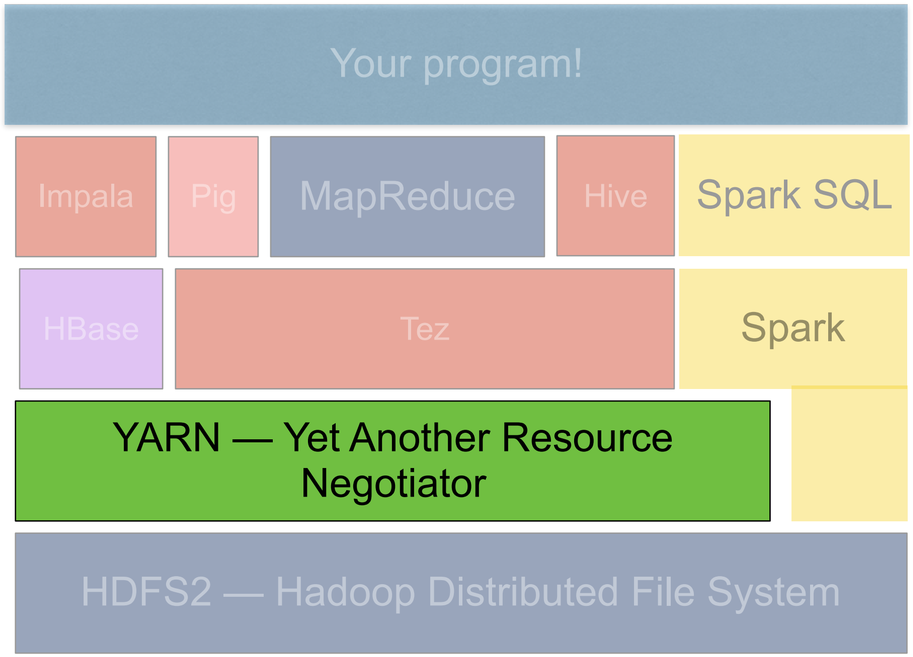




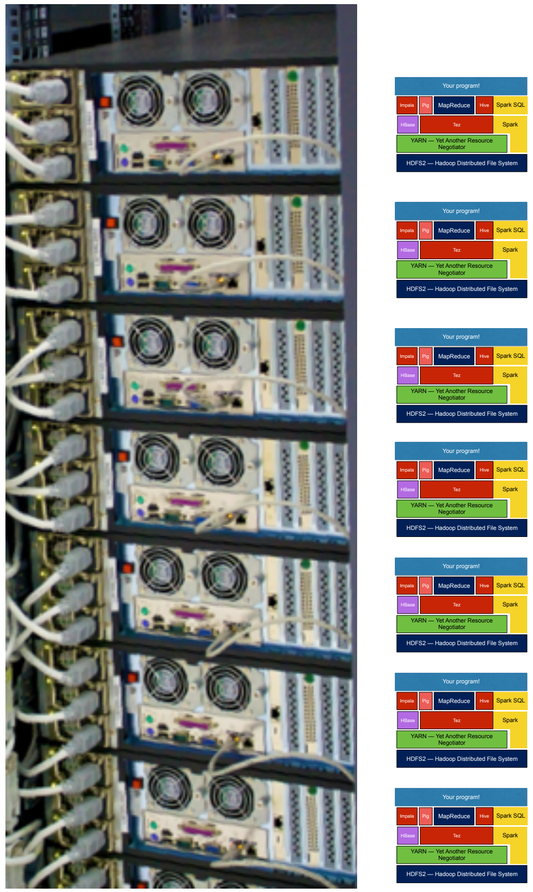
YARN — Yet Another Resource Negotiator
Manages the resources for the entire cluster.

Receives submitted jobs.

Allocates nodes to tasks

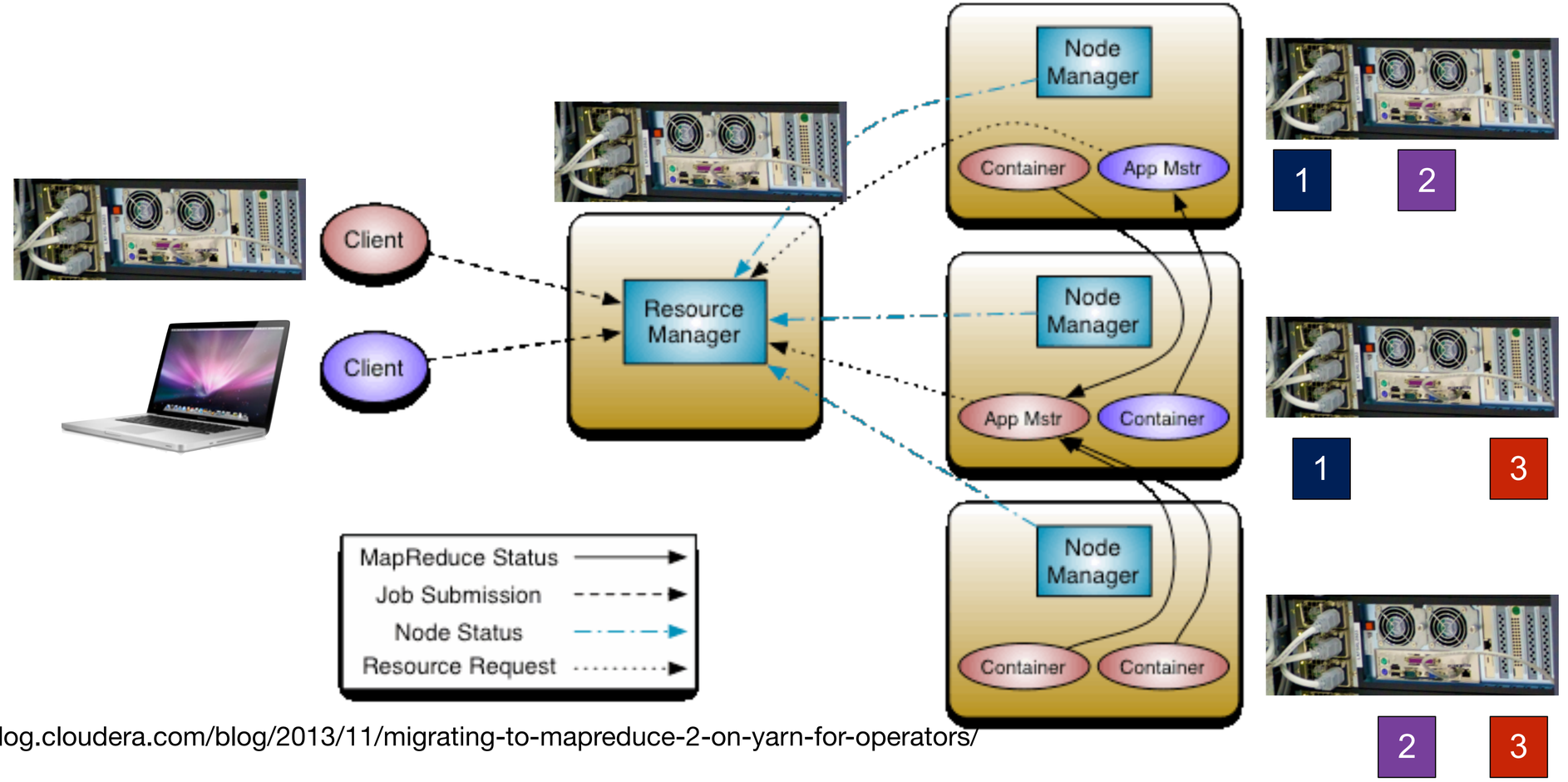


Rack



Yarn receives jobs from "clients" and sends them to the nodes.

Tasks are sent to the nodes that have the data (if possible)



- http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/

YARN is a persistant Java program and acommand-line utility.

/usr/bin/yarn — bash script that runs /usr/lib/hadoop-yarn/bin/yarn

/usr/lib/hadoop-yarn/bin/yarn — bash script that runs the appropriate Java class

- https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

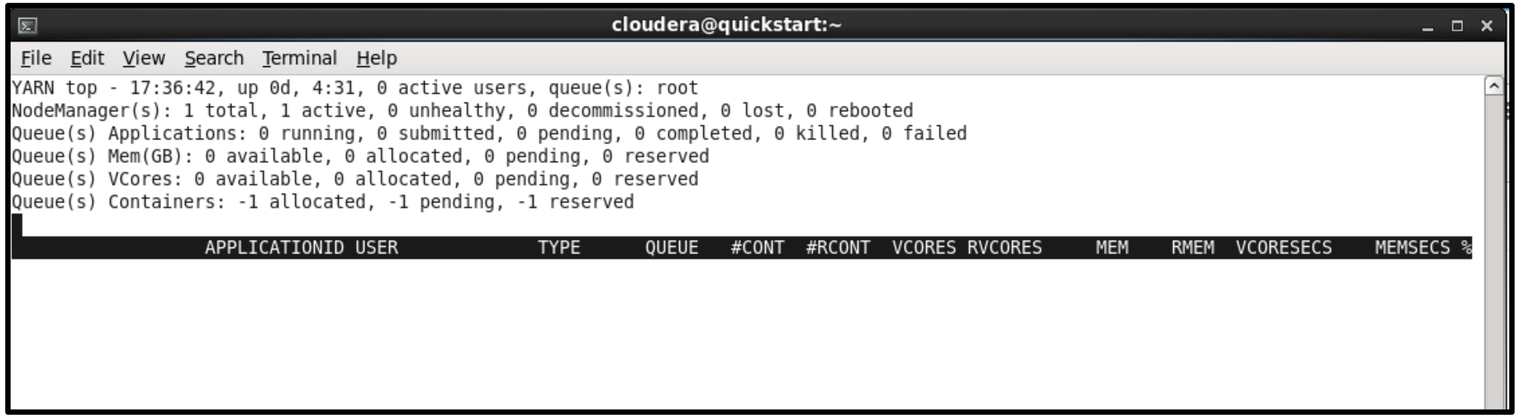
```
[cloudera@quickstart ~]$ yarn
Usage: yarn [--config confdir] COMMAND
where COMMAND is one of:
  resourcemanager -format-state-store   deletes the RMStateStore
  resourcemanager                       run the ResourceManager
  nodemanager                           run a nodemanager on each slave
  timelineserver                        run the timeline server
  rmadmin                               admin tools
  version                               print the version
  jar <jar>                             run a jar file
  application                           prints application(s) report/kill application
  applicationattempt                    prints applicationattempt(s) report
  container                             prints container(s) report
  node                                  prints node report(s)
  queue                                 prints queue information
  logs                                  dump container logs
  classpath                             prints the class path needed to get the Hadoop jar and the required libraries
  daemonlog                             get/set the log level for each daemon
  top                                   run cluster usage tool
 or
  CLASSNAME                             run the class named CLASSNAME

Most commands print help when invoked w/o parameters.
[cloudera@quickstart ~]$
```

Basic Yarn commands

“yarn jar” — Submits a Hadoop job:

```
$ export STREAMING_JAR=/usr/lib/hadoop-mapreduce/hadoop-streaming.jar
$ yarn jar $STREAMING_JAR -mapper mapper.py -reducer reducer.py \
  -input /user/myusername/input -output /user/myusername/gutenberg_output -file mapper.py -file reducer.py
```

$ yarn top — shows running jobs



- https://hadoop.apache.org/docs/current/hadoop-streaming/HadoopStreaming.html

More commands…

$ yarn node -list -all — List all nodes

```
[cloudera@quickstart ~]$ yarn node -list -all
16/01/10 17:37:34 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
Total Nodes:1
         Node-Id             Node-State      Node-Http-Address     Number-of-Running-Containers
quickstart.cloudera:44490         RUNNING      quickstart.cloudera:8042                    0
[cloudera@quickstart ~]$
```

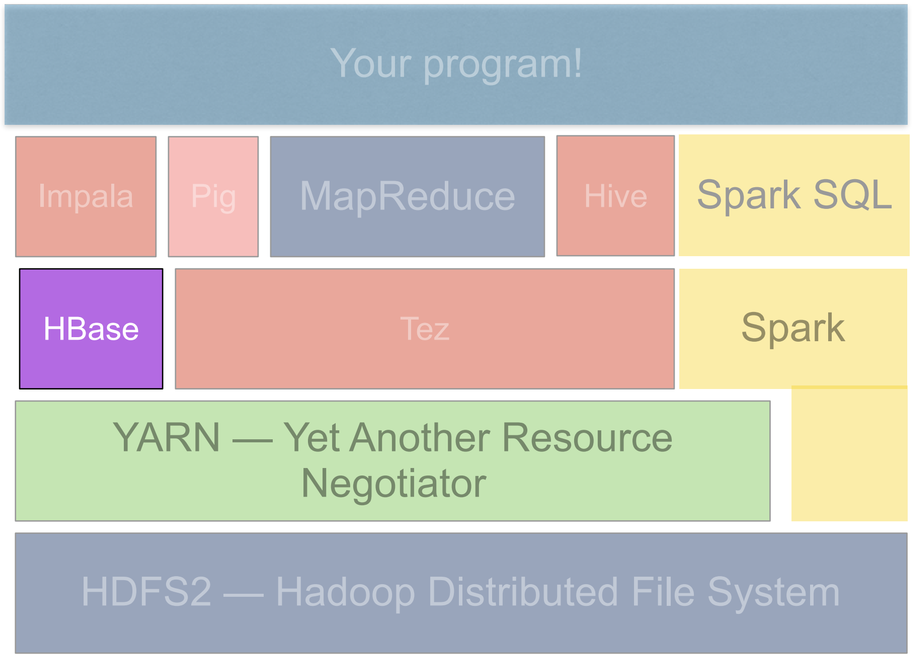
Yarn is a full cluster management system.

- http://www.slideshare.net/cloudera/introduction-to-yarn-and-mapreduce-2

HBase layer manages a distributed database.

HBase:

- Column-oriented database, stores data in HDFS
- Key-value store
- Based on Google's "Big Table"
- Not an RDBMS



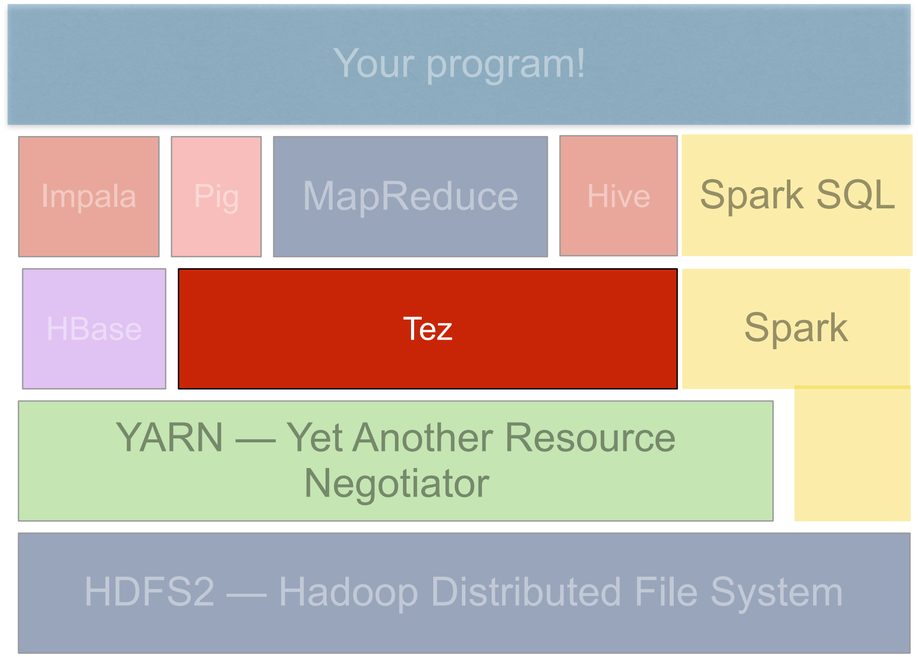
Data

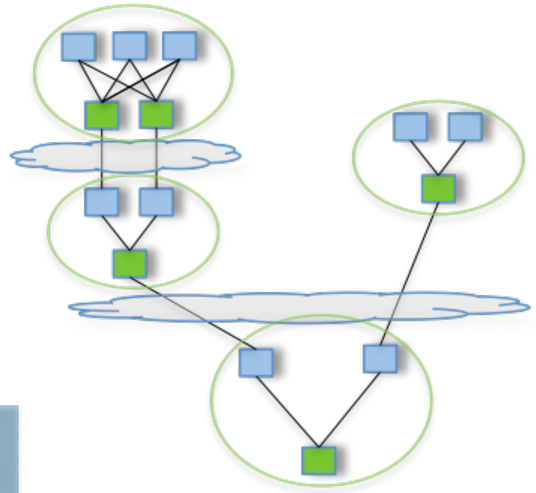
Tez* performs data-flow analysis using "directed-acyclic-graphs"

Removes redundant jobs.

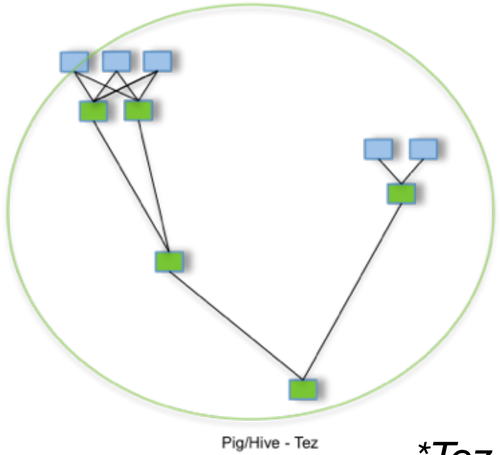
Manages workloads

- https://tez.apache.org/
- http://hortonworks.com/hadoop/tez/





Pig/Hive - MR



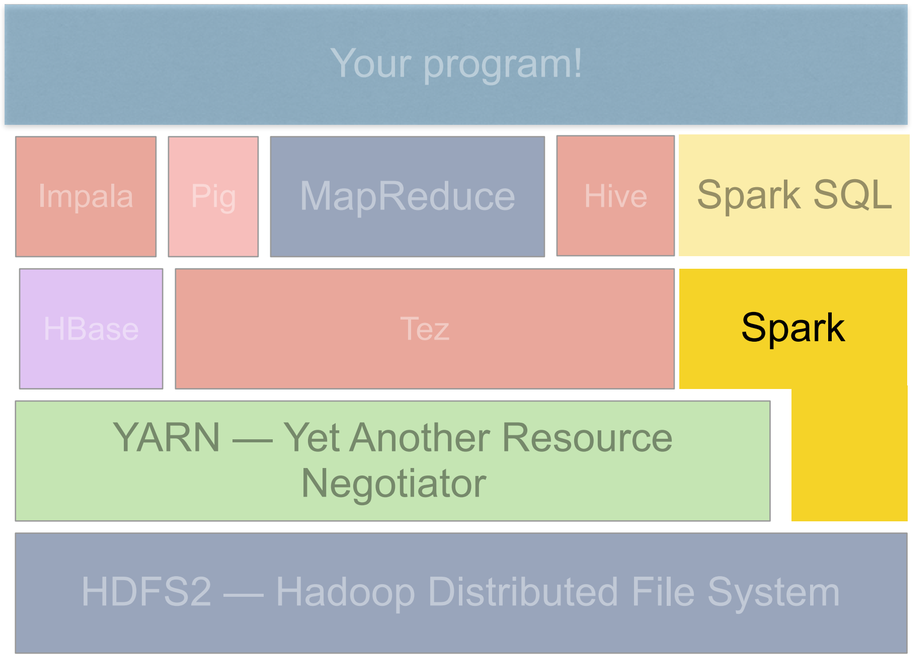
Pig/Hive - Tez

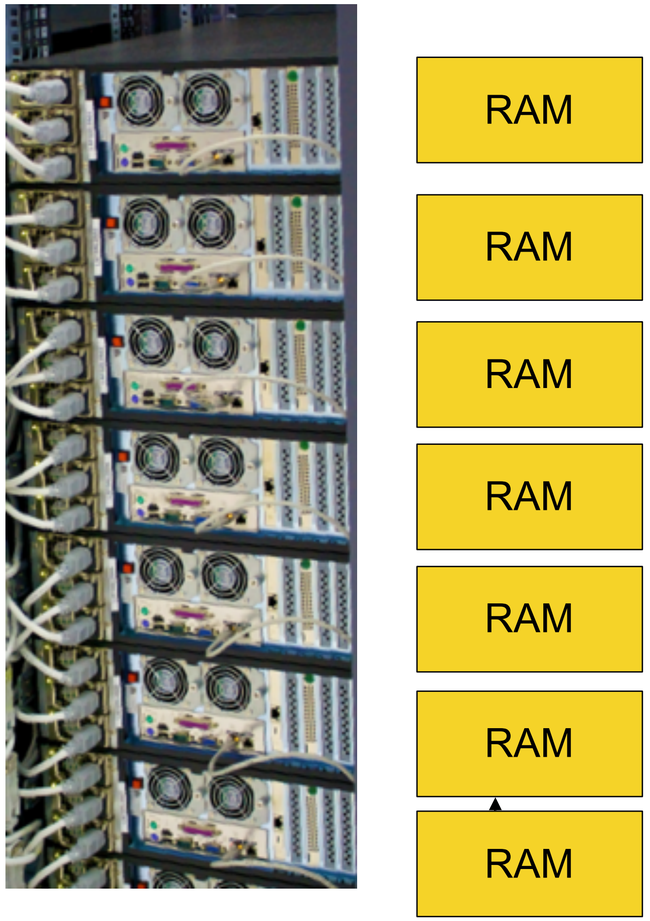
Tez is hindi for "fast."

Spark performs calculations in RAM

Spark:

- Data moves RAM→RAM without being written to disk.
- 5x - 50x faster than MapReduce!
- Easier to program!
- Can run on YARN or on bare metal.

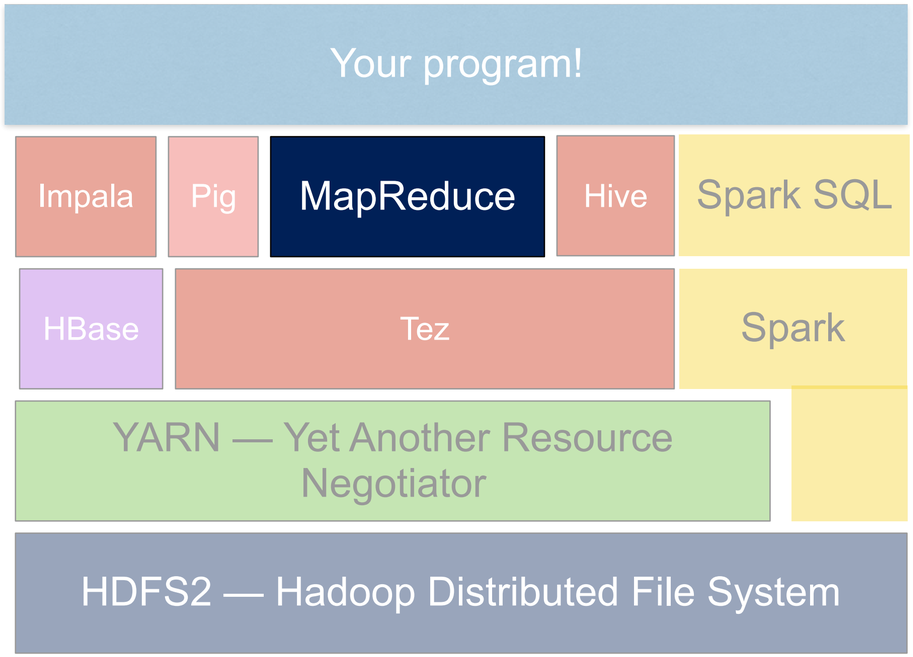




MapReduce — Handles Map/Combiner/Partition/Shuffle/Reduce

Hooks for calling Java programs as mappers, combiners, partitioners, & reducers

Hadoop streaming run as a special Java class.



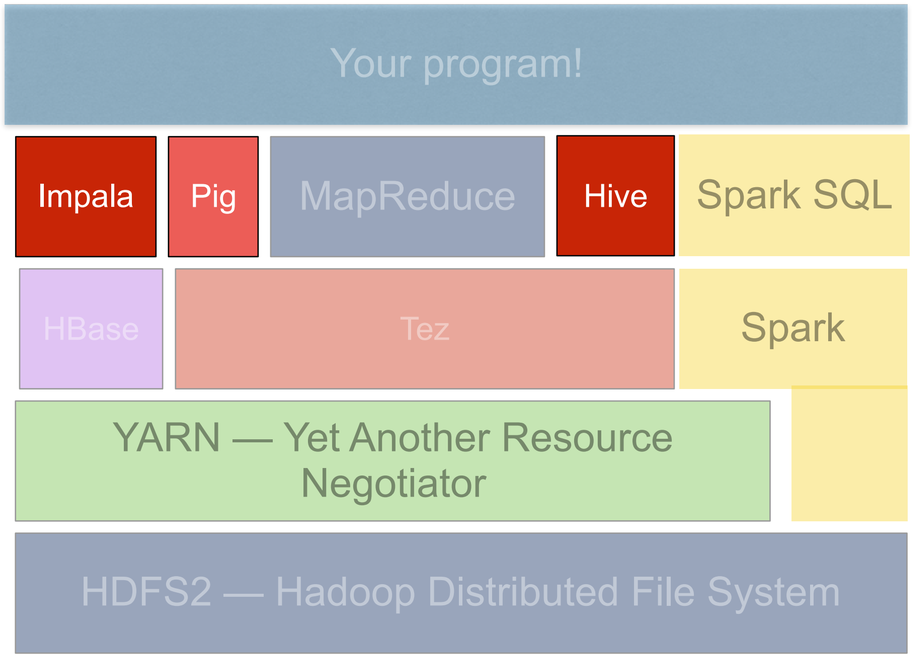
Impala, Pig, Hive, Sqoop & Spark SQL: SQL-interfaces for Hadoop

Impala — transfers between Hadoop and relational databases.

Hive — manages large datasets in HDFS

Pig — Compiles SQL-like queries
in “Pig Latin” to MapReduce jobs

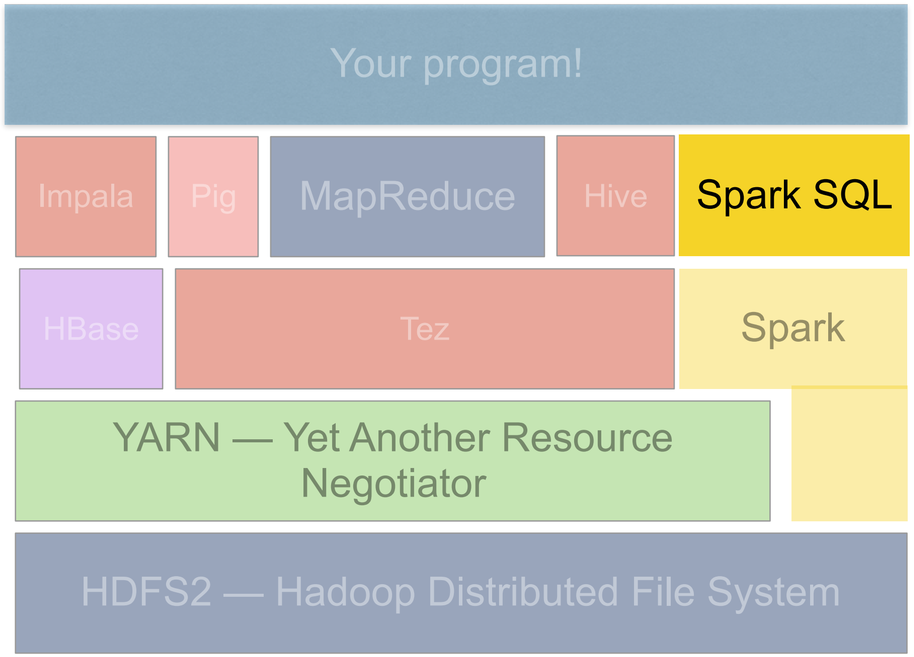
Sqoop — batch interface between SQL and HDFS



SparkSQL

Another SQL-like interface to Spark, HBase, & HDFS

Well integrated with Spark.



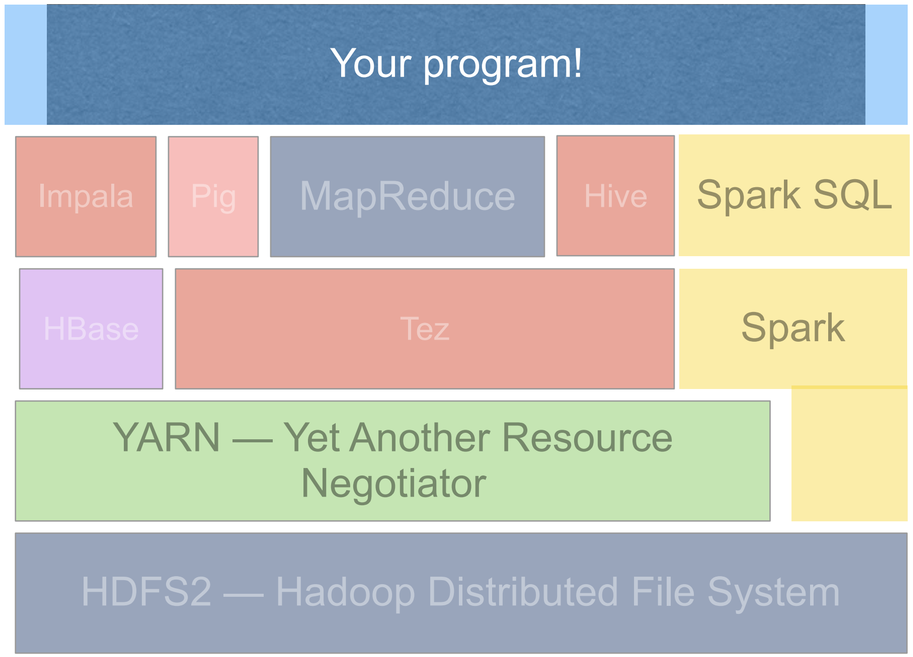
Your program!

Most jobs consist of multiple operations:

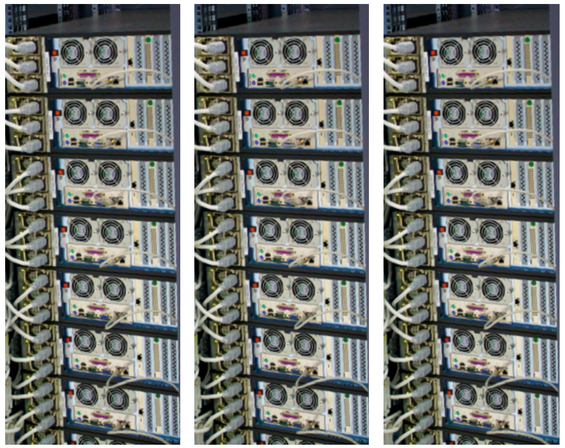
- Fetch data from Database/ HDFS / S3
- Process (multiple MapReduce or Spark steps)
- Save data in HDFS or S3 (if large) or to STDOUT (if small)

Because you use Hadoop...

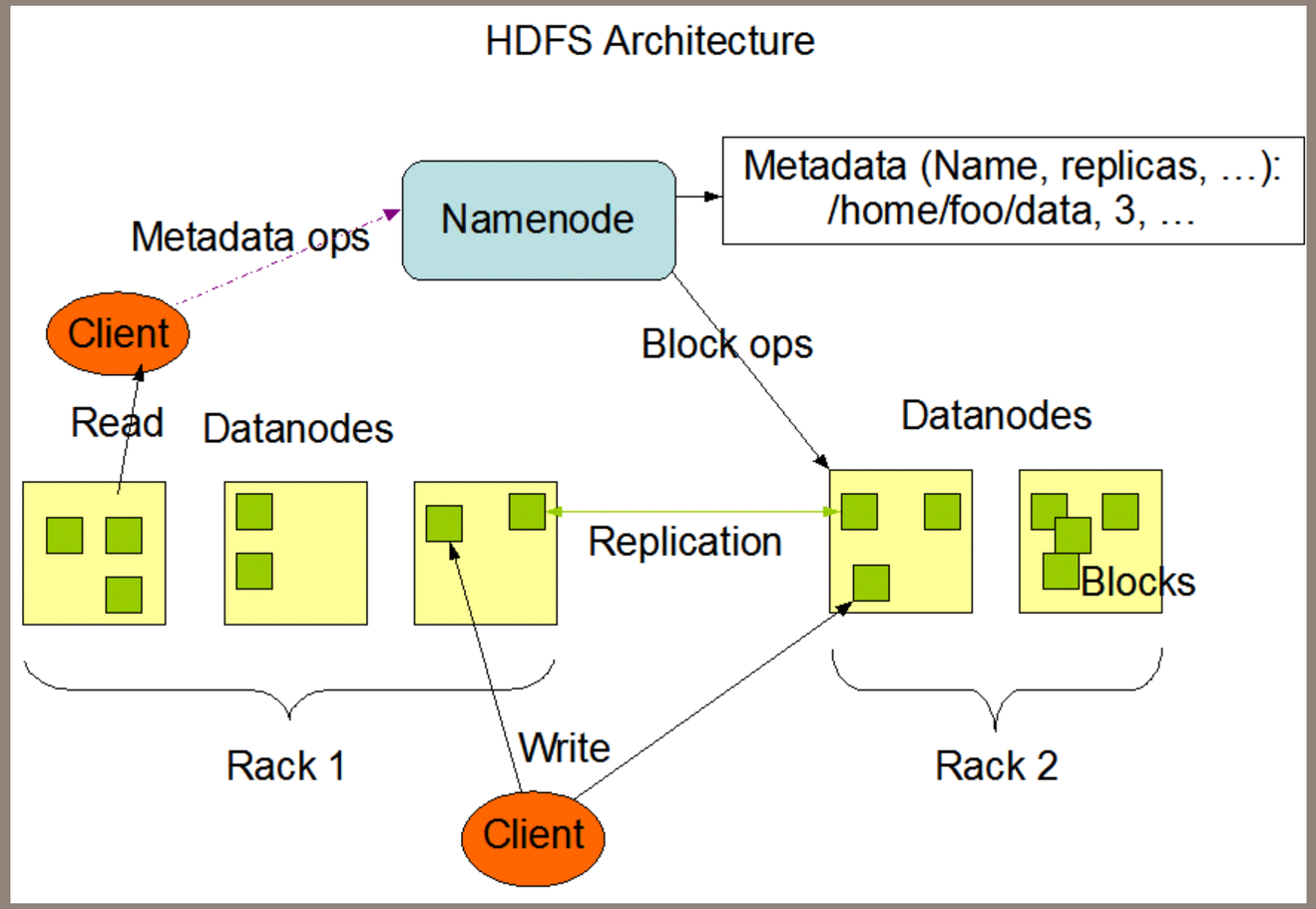
- 3x computers makes the job run in 1/3 the time



6 hours



2 hours



HDFS: The Hadoop Distributed File System

HDFS Design Considerations

Commodity hardware

- No special RAID controllers, high-performance interconnect, etc.

Scale to thousands of nodes.

High aggregate throughput

- One node: 10 min to read 60GB (100 MBB/sec)
- 100 nodes: 10 min to read 6TB (10,000 MB/sec)
  - *—Goal is to read 6TB in 10 min, not to read 60GB in 10 seconds*

Designed for batch processing of data:

- Data stored in sequential files.
- Files can be read, written, or appended.
  - *—Data cannot be changed after written.*

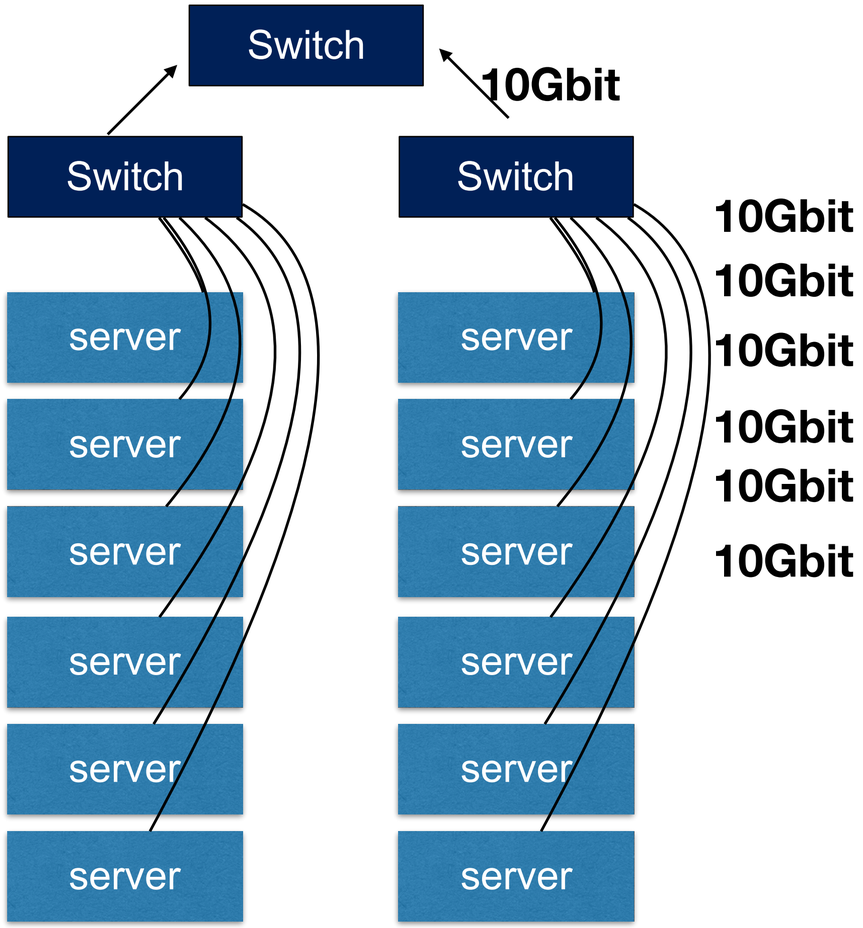
Data Replication — handles failures and parallelism

- If a drive or server fails, clients read from another server.

Rack-aware architecture: Computers are deployed in racks

Rack-aware architecture

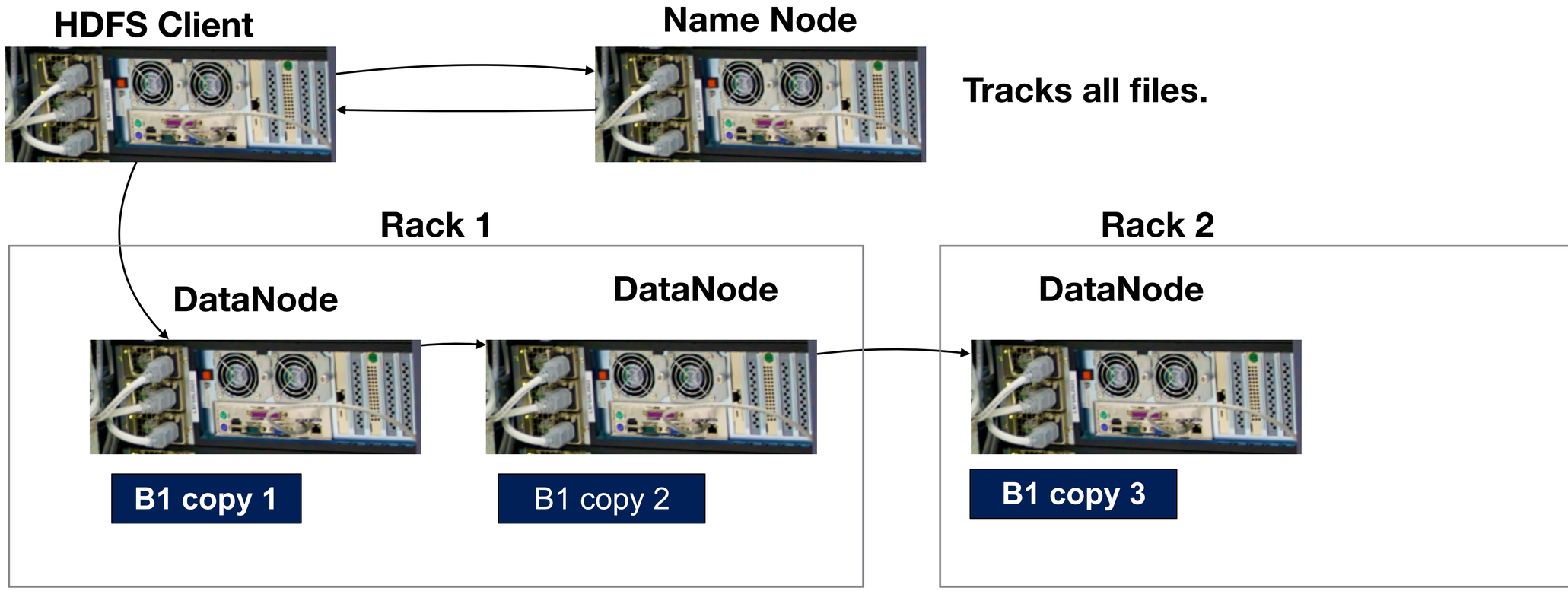
- More bandwidth within racks than between racks



HDFS Architecture: Writing

Each file => 1 or more blocks

Each block => 1 or more nodes (replication factor)



Heartbeat — Resiliency

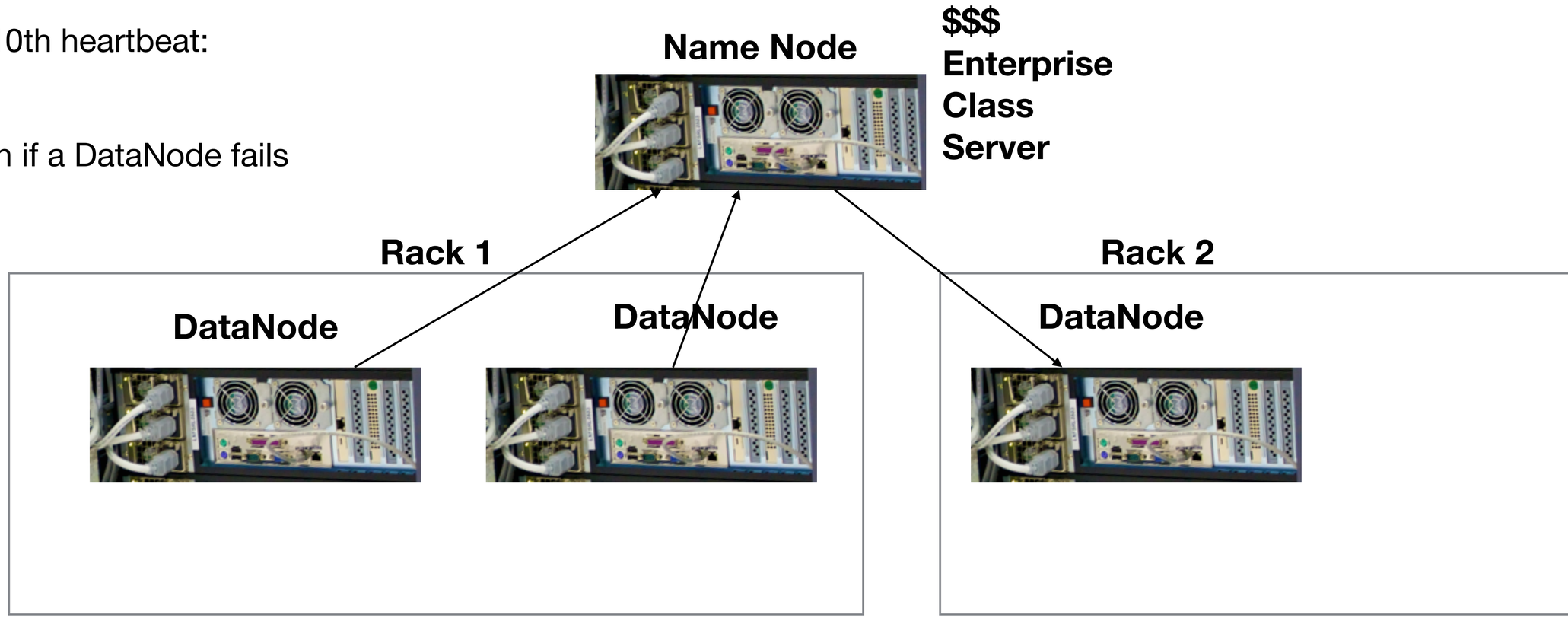
dfs.heartbeat.interval — default 3 seconds

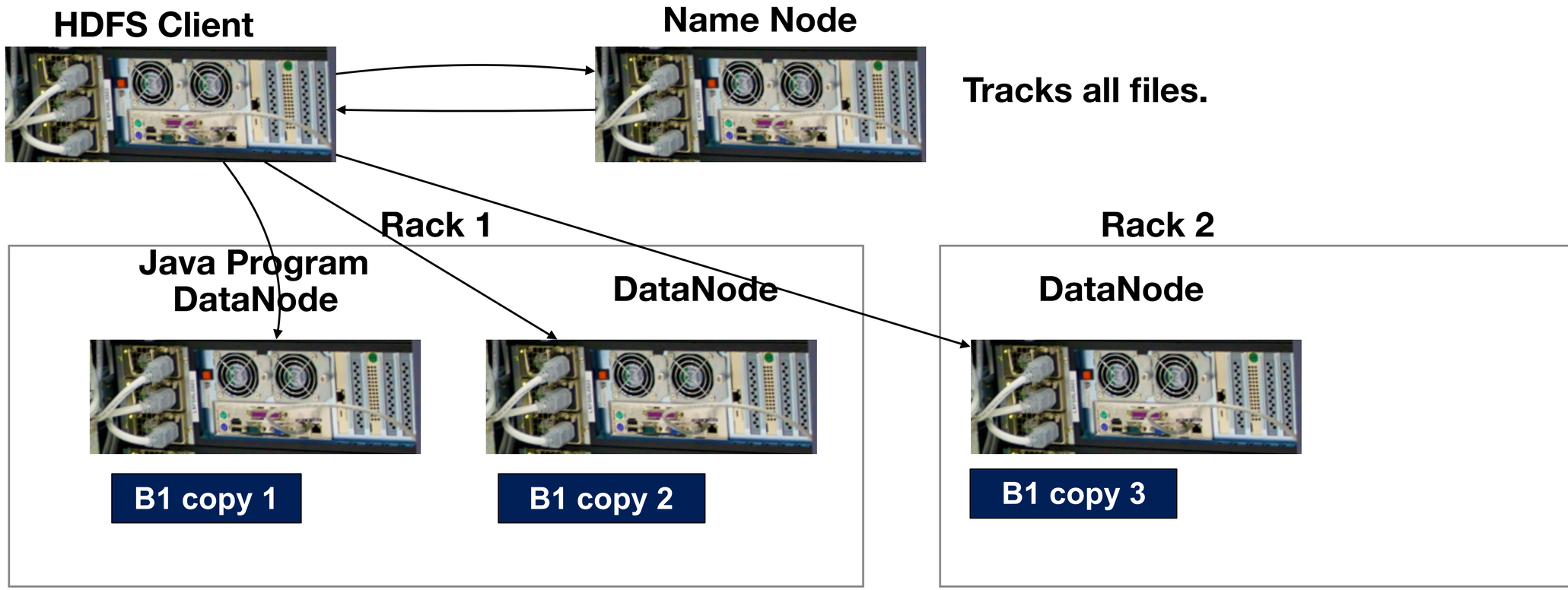
Reports:

- “I’m alive!”
- Block Report every 10th heartbeat:

Name Node:

- Schedules replication if a DataNode fails




HDFS Client
Name Node
Tracks all files.
Rack 1
Rack 2
Java Program
DataNode
DataNode
DataNode
B1 copy 1
B1 copy 2
B1 copy 3

Replication — Determines Performance and Overhead

Replication = 1 (no replication)

- Cloudera VM
- Amazon EMR with 1 data node
- No backup against failure.

Replication = 3 (typical)*

- Each block stored 3 times
- Name Node keeps track which blocks on which servers
  - *—If a server fails, Name Node tells replicants to make a copy.*

Block Size — default is 64MiB

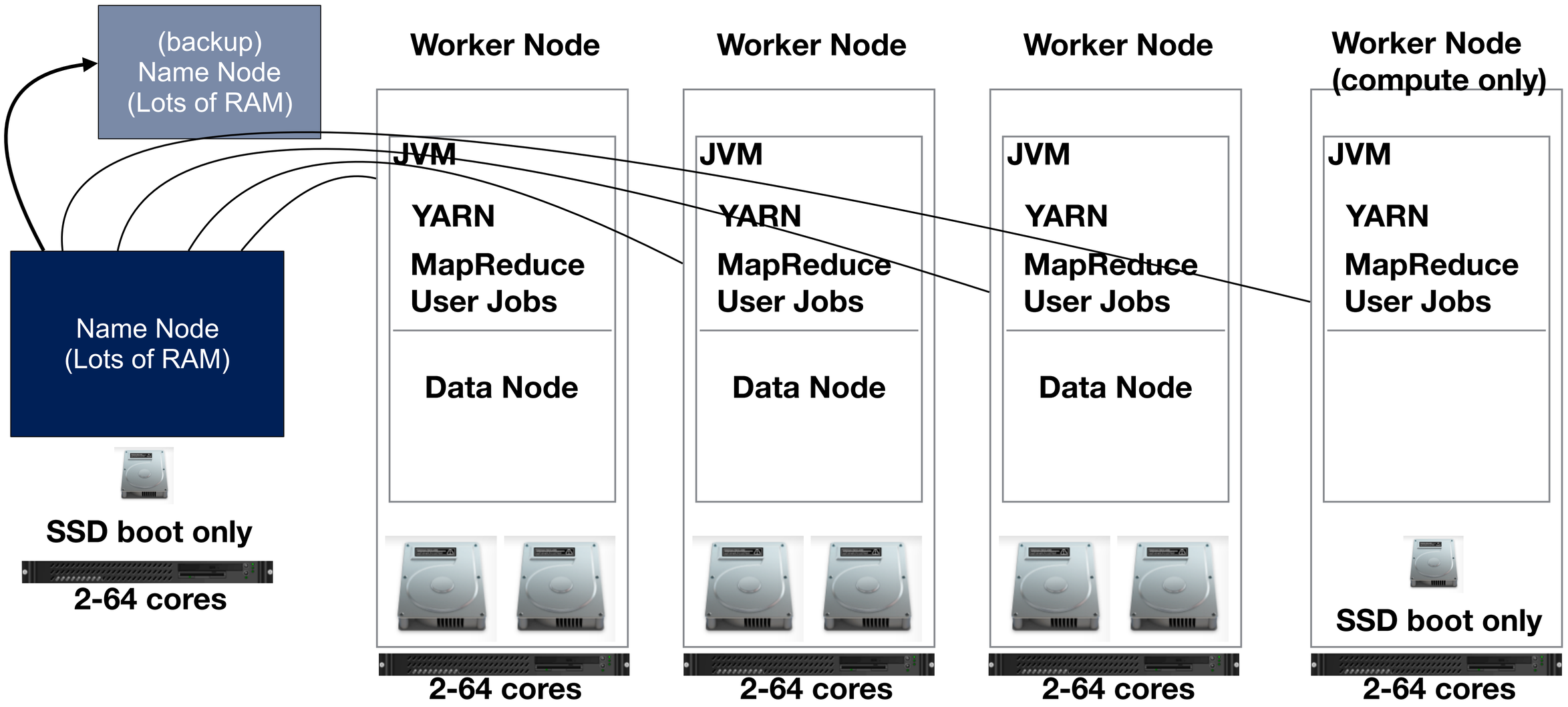
- Don't store small files!
- Hadoop "Sequence File" stores lots of small files in a single "file."

∴ Storing a 1GiB file with replication 3 takes 3GiB (1024 ÷ 64 x 3 = 48 blocks)

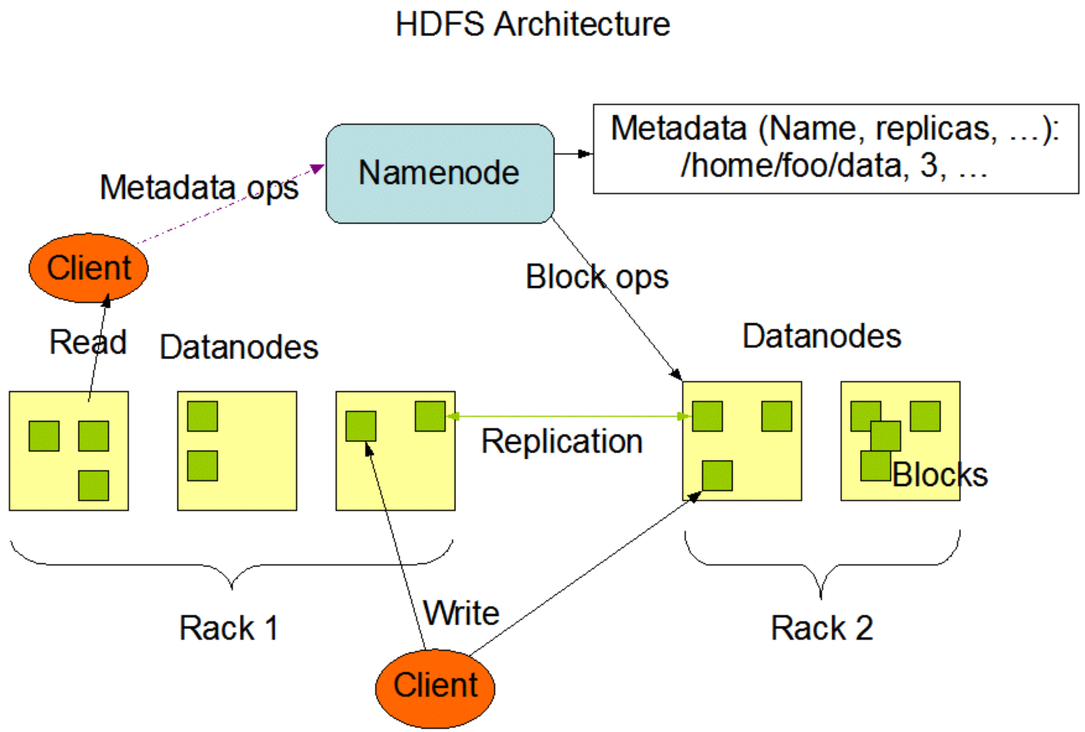
∴ Storing 100 1MiB files with replication 3 takes 19.2GiB (100 x 3 = 300 blocks)

- * HDFS3 will use erasure coding instead of replication for low I/O files.
  http://blog.cloudera.com/blog/2015/09/introduction-to-hdfs-erasure-coding-in-apache-hadoop/

Remember, each "worker node" is potentially both a data node and a compute node.



Apache's "official" HDFS architecture diagram



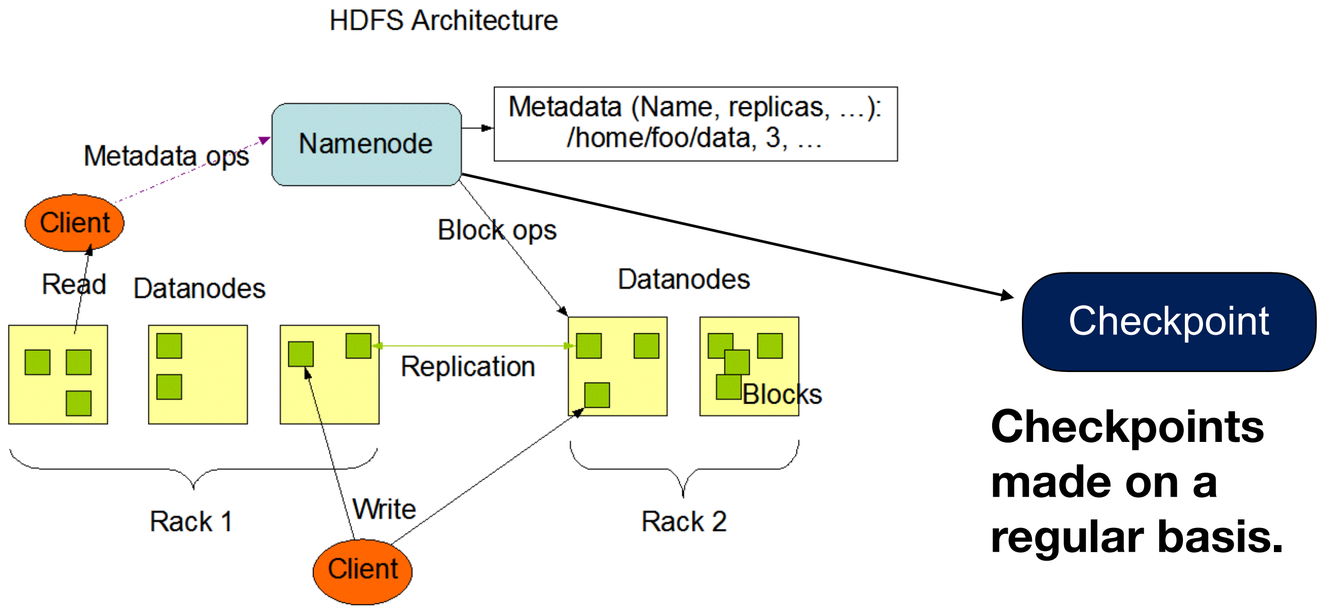
NameNode scaling issues: The Small Files Problem

NameNode keeps entire file system in memory!

Each file requires:

- A file inode reference (≈150 bytes) + a block reference (≈150 bytes) = ≈300 bytes
- A million files — 300MB (a typical laptop has
- A billion files — 300GB
- Ten billion files? — 3TB?

—*Not happening!*



Checkpoints made on a regular basis.

https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Other problems with small files

More files means:

- More wasted space (64MB block size)
- More map tasks (each file needs its own map task)
- Map tasks are largely wasted.

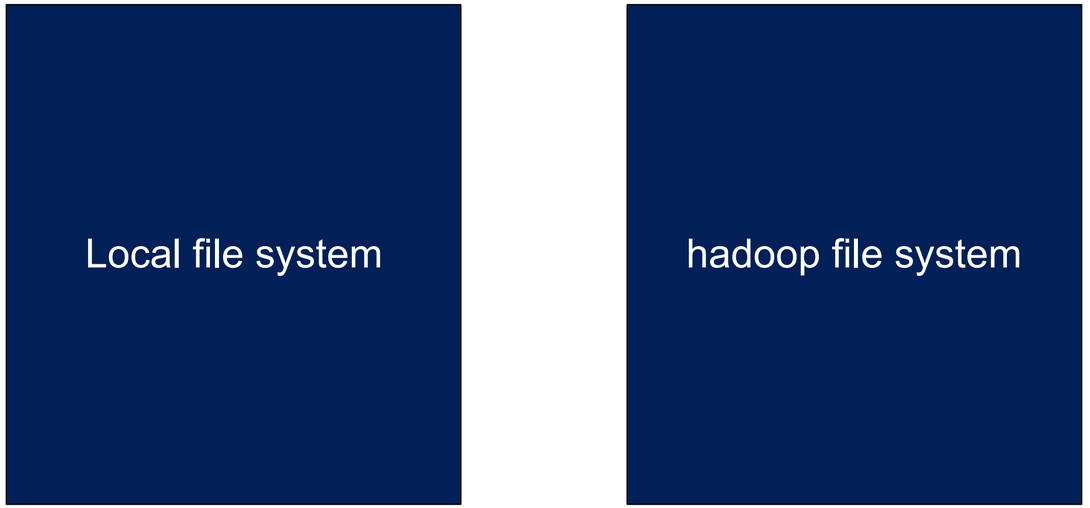
Therefore — Keep your files big!

- Combine many small files into a few big files:
  - —Hadoop SequenceFiles — splittable, compressible, for working with large amounts of binary data. Java Only.
  - —Hadoop MapFiles — Indexed sequence files.
  - —Hadoop Archive Files
- HBase — Database abstraction from a few large files
- S3 — Easily handles lots of files

If your total data isn't >500MB, you probably shouldn't be using Hadoop!

Interacting with HDFS: “hdfs dfs” command

“hdfs” is the primary command for interacting with the Hadoop file system



cat	**hdfs dfs -cat**
cp	**hdfs dfs -cp**
df	**hdfs dfs -df**
du	**hdfs dfs -du**

- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html

Make a remote directory; copy files; list them; cat them

```
[cloudera@quickstart ~]$ hdfs dfs -rm -R tmp
rm: `tmp': No such file or directory
[cloudera@quickstart ~]$ cat README.md
This is a readme file.
one
two
three.
Just another file

[cloudera@quickstart ~]$ hdfs dfs -ls tmp
ls: `tmp': No such file or directory
[cloudera@quickstart ~]$ hdfs dfs -mkdir tmp
[cloudera@quickstart ~]$ hdfs dfs -put README.md tmp/
[cloudera@quickstart ~]$ hdfs dfs -ls tmp/
Found 1 items
-rw-r--r--   1 cloudera cloudera         58 2015-12-10 19:41 tmp/README.md
[cloudera@quickstart ~]$ hdfs dfs -cat tmp/README.md
This is a readme file.
one
two
three.
Just another file

[cloudera@quickstart ~]$
```

Informational Commands

```
[cloudera@quickstart ~]$ hdfs dfs -du
58    58    README.md
58    58    demo
1453  1453  join1_mapper.py
3951  3951  join1_reducer.py
1186  1186  make_join2data.py
94    94    output_new
94    94    output_new_2
1007  1007  outz
70    70    streaming-input
94    94    streaming-output
58    58    tmp
60    60    wordcount
[cloudera@quickstart ~]$ hdfs dfs -df
Filesystem                                Size       Used    Available  Use%
hdfs://quickstart.cloudera:8020  58531520512  438538240  44233338880    1%
[cloudera@quickstart ~]$
```

hdfs dfsadmin — administrative commands

```
[cloudera@quickstart ~]$ hdfs dfsadmin -report
Configured Capacity: 58531520512 (54.51 GB)
Present Capacity: 44671877605 (41.60 GB)
DFS Remaining: 44237303808 (41.20 GB)
DFS Used: 434573797 (414.44 MB)
DFS Used%: 0.97%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (1):

Name: 127.0.0.1:50010 (quickstart.cloudera)
Hostname: quickstart.cloudera
Decommission Status : Normal
Configured Capacity: 58531520512 (54.51 GB)
DFS Used: 434573797 (414.44 MB)
Non DFS Used: 13859642907 (12.91 GB)
DFS Remaining: 44237303808 (41.20 GB)
DFS Used%: 0.74%
DFS Remaining%: 75.58%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 6
Last contact: Thu Dec 10 19:45:28 PST 2015


[cloudera@quickstart ~]$
```

Tuning HDFS for performance and robustness

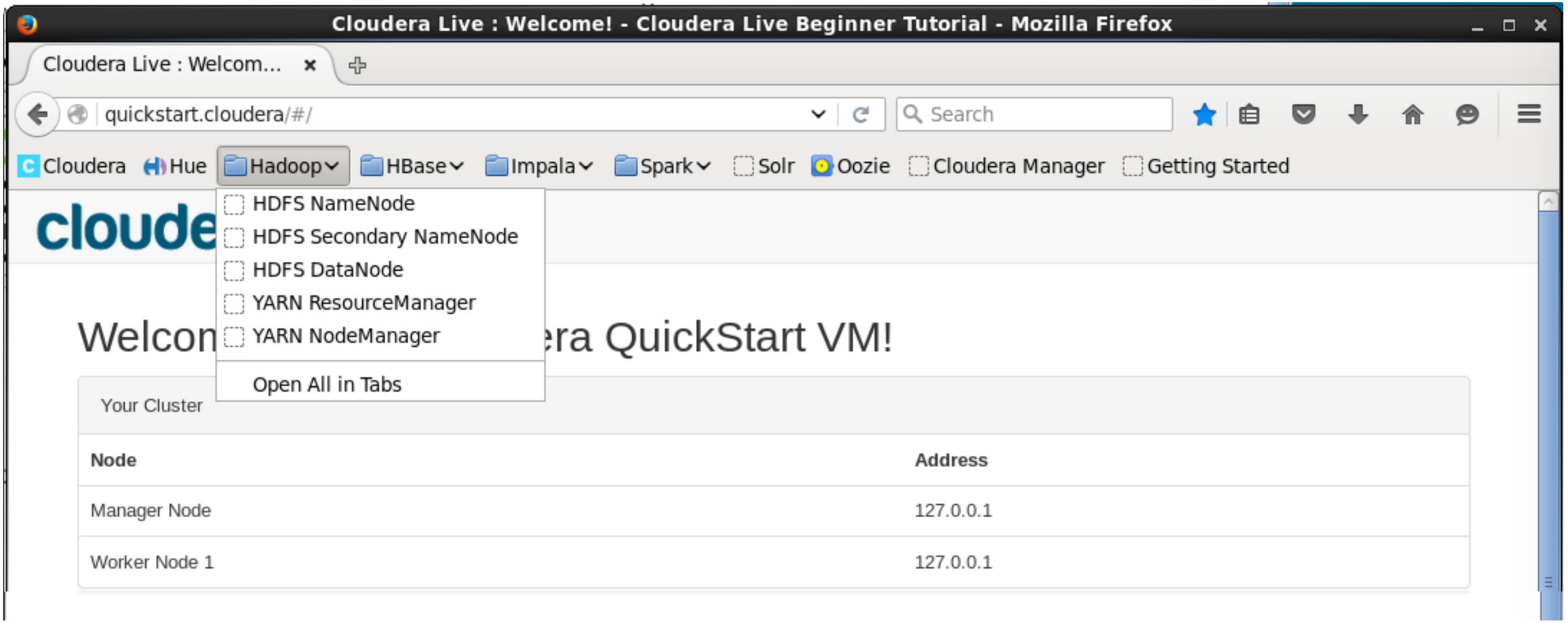
File system configuration parameters: /etc/hadoop/conf/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <!-- Immediately exit safemode as soon as one DataNode checks in.
       On a multi-node cluster, these configurations must be removed.  -->
  <property>
    <name>dfs.safemode.extension</name>
    <value>0</value>
  </property>
  <property>
     <name>dfs.safemode.min.datanodes</name>
     <value>1</value>
  </property>
  <property>
     <name>dfs.safemode.min.datanodes</name>
     <value>1</value>
  </property>
  <property>
     <name>dfs.permissions.enabled</name>
     <value>false</value>
  </property>
  <property>
     <name>dfs.permissions</name>
     <value>false</value>
  </property>
```

- https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

—*"You can tune a file system, but you can tune a fish." — UNIX Man Page, circa*

The quickstart VM has an HDFS browser



Security is off.

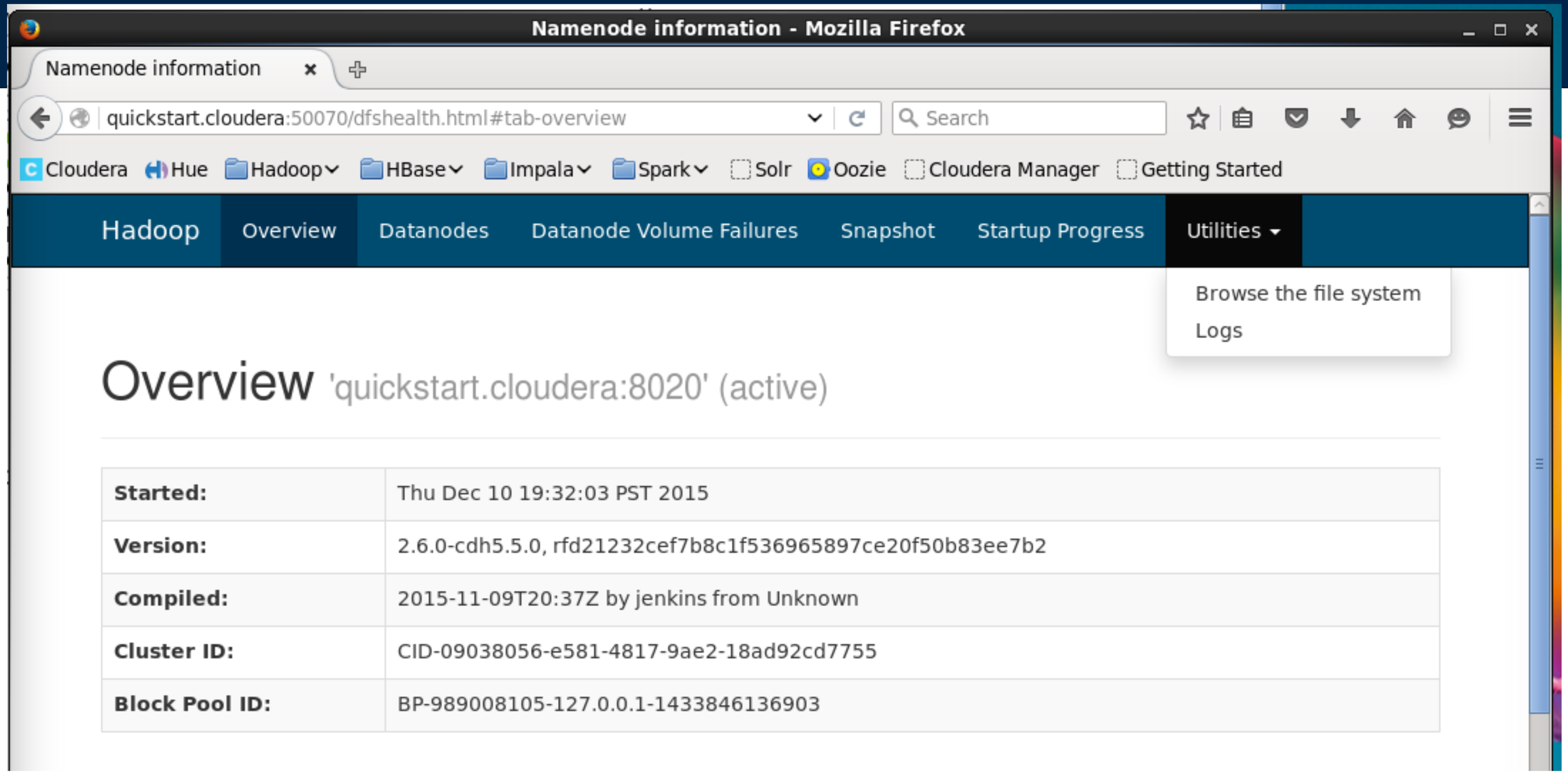
Safemode is off.

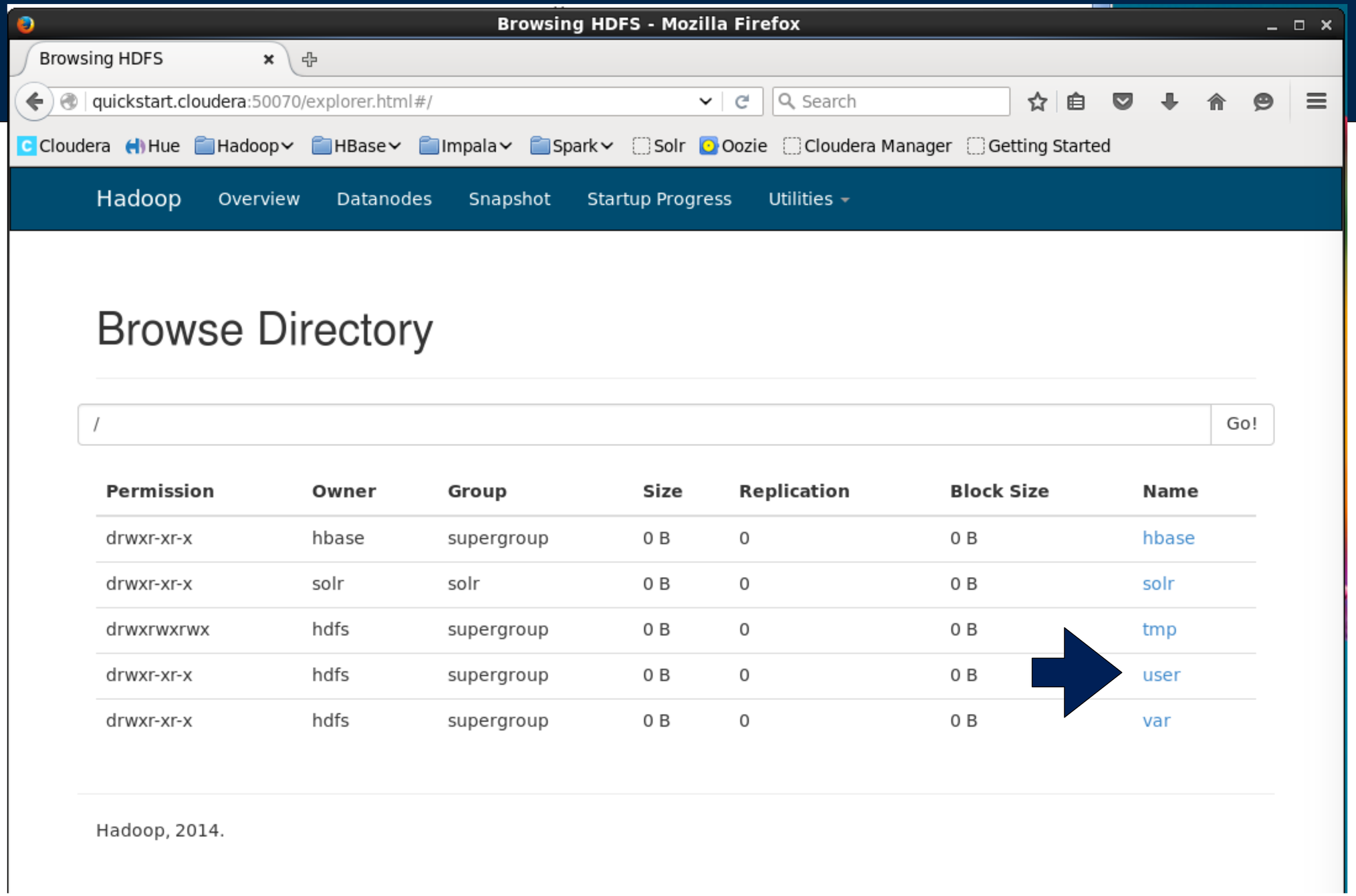
802 files and directories, 565 blocks = 1,367 total filesystem object(s).

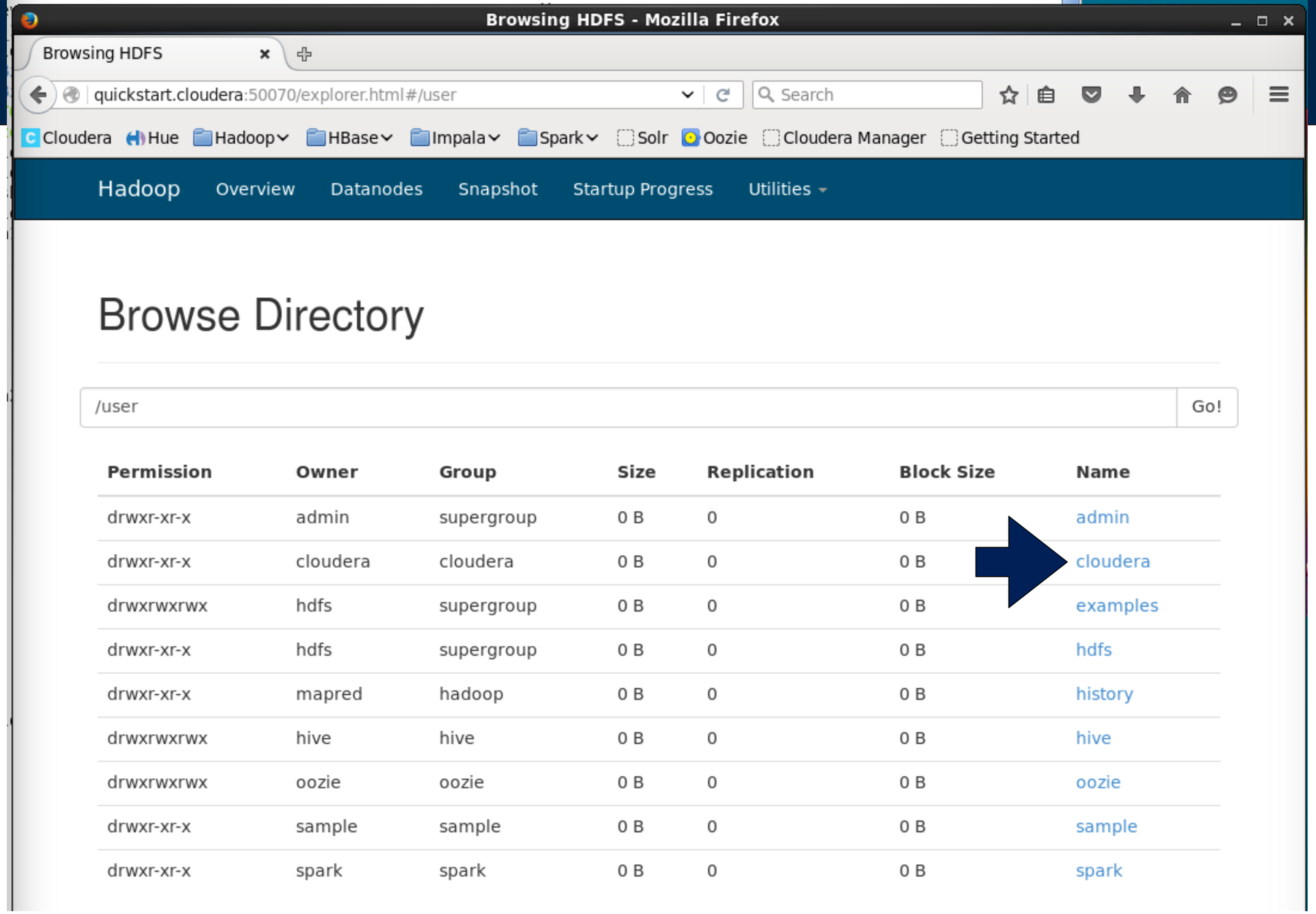
Heap Memory used 106.5 MB of 211 MB Heap Memory. Max Heap Memory is 889 MB.

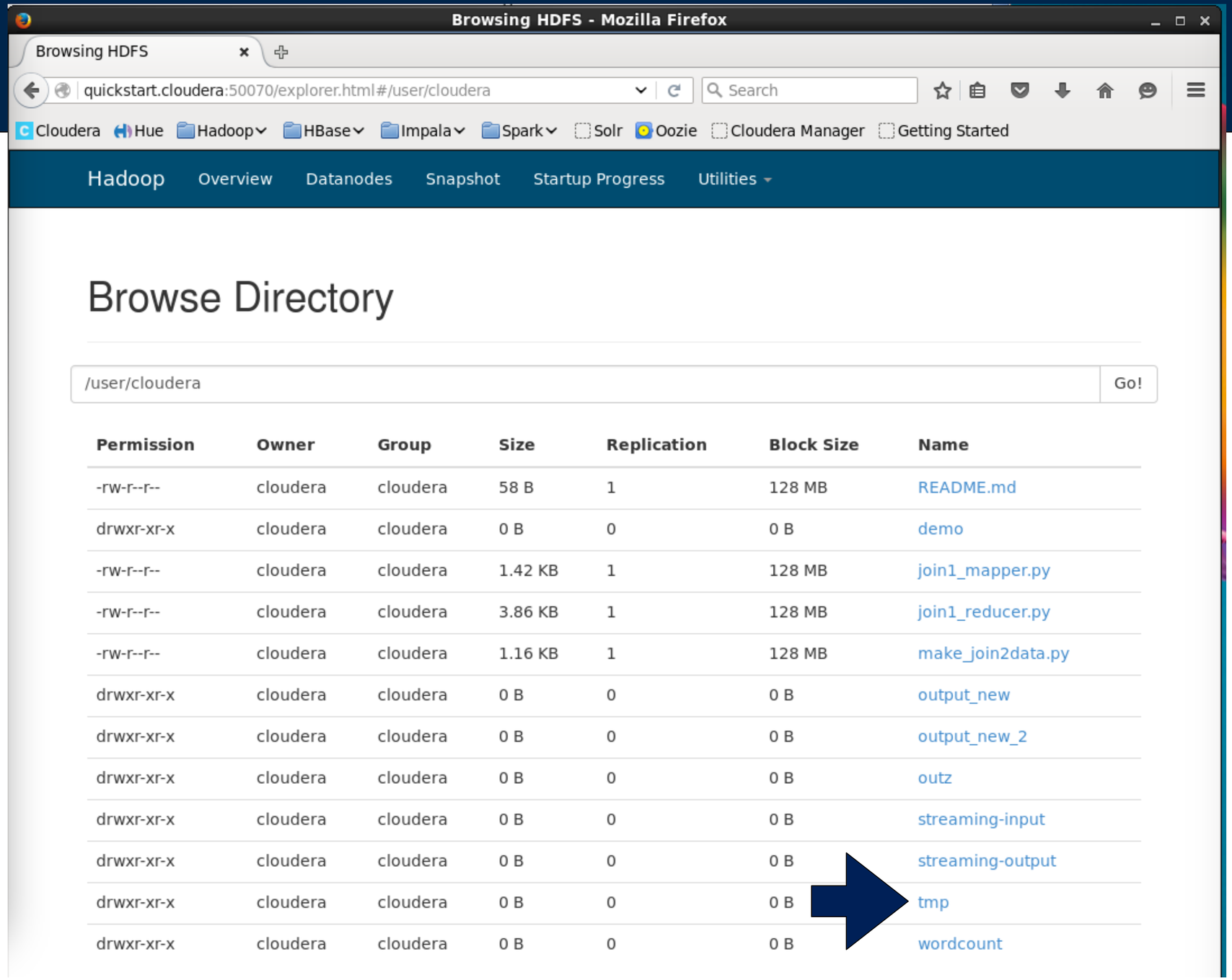
Non Heap Memory used 33.71 MB of 34.94 MB Commited Non Heap Memory. Max Non Heap Memory is 130 MB.

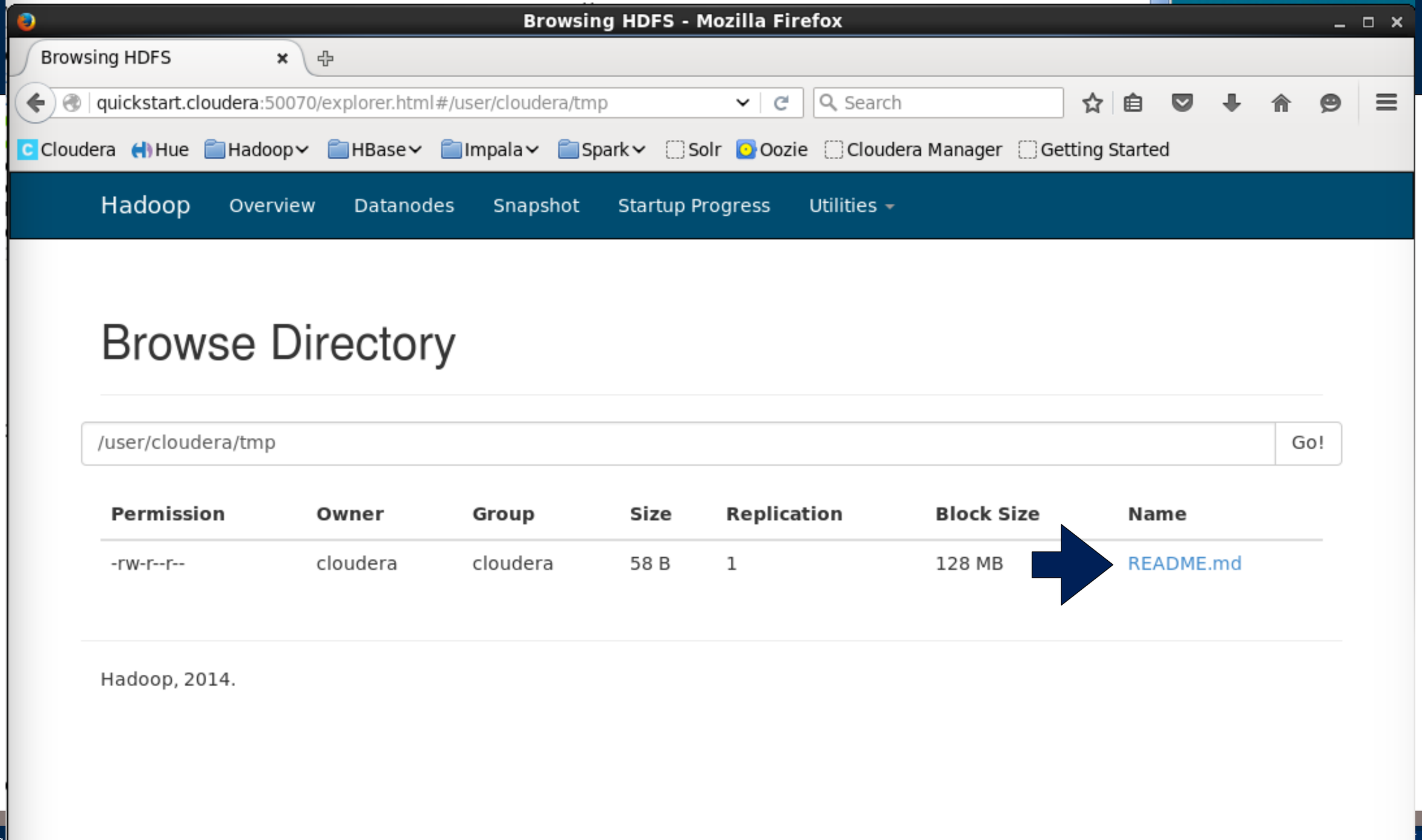
Configured Capacity:	54.51 GB
DFS Used:	414.44 MB
Non DFS Used:	12.91 GB
DFS Remaining:	41.2 GB
DFS Used%:	0.74%
DFS Remaining%:	75.58%
Block Pool Used:	414.44 MB
Block Pool Used%:	0.74%
DataNodes usages% (Min/Median/Max/stdDev):	0.74% / 0.74% / 0.74% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)

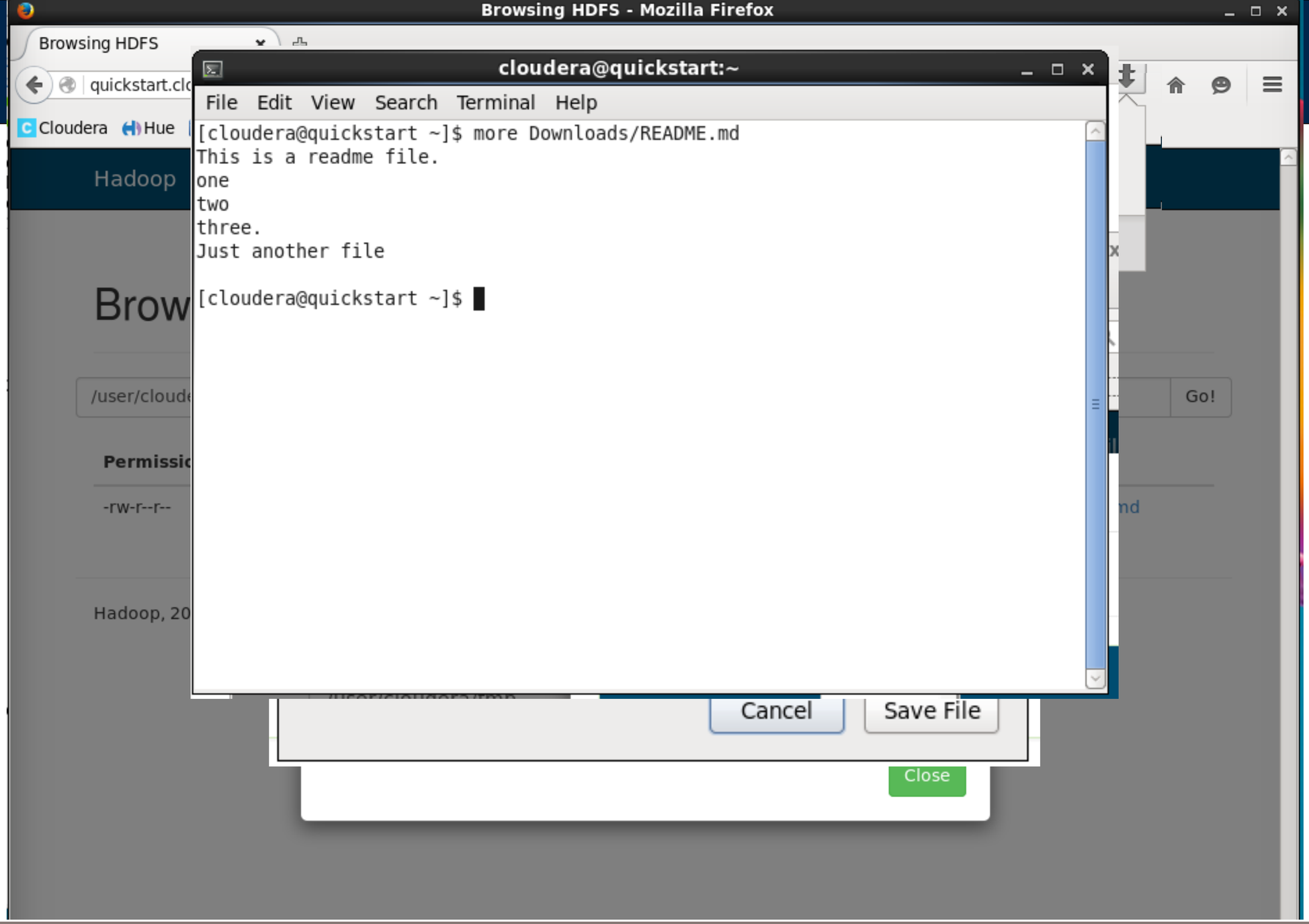

Namenode information - Mozilla Firefox
Namenode information
quickstart.cloudera:50070/dfshealth.html#tab-overview
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse the file system
Logs
Overview 'quickstart.cloudera:8020' (active)
Started: Thu Dec 10 19:32:03 PST 2015
Version: 2.6.0-cdh5.5.0, rfd21232cef7b8c1f536965897ce20f50b83ee7b2
Compiled: 2015-11-09T20:37Z by jenkins from Unknown
Cluster ID: CID-09038056-e581-4817-9ae2-18ad92cd7755
Block Pool ID: BP-989008105-127.0.0.1-1433846136903


Browsing HDFS - Mozilla Firefox
Browsing HDFS
quickstart.cloudera:50070/explorer.html#/
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Permission
Owner
Group
Size
Replication
Block Size
Name
drwxr-xr-x
hbase
supergroup
0 B
0
0 B
hbase
drwxr-xr-x
solr
solr
0 B
0
0 B
solr
drwxrwxrwx
hdfs
supergroup
0 B
0
0 B
tmp
drwxr-xr-x
hdfs
supergroup
0 B
0
0 B
user
drwxr-xr-x
hdfs
supergroup
0 B
0
0 B
var
Hadoop, 2014.


Browsing HDFS - Mozilla Firefox
Browsing HDFS
quickstart.cloudera:50070/explorer.html#/user
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/user
Go!
Permission
Owner
Group
Size
Replication
Block Size
Name
drwxr-xr-x admin supergroup 0 B 0 0 B admin
drwxr-xr-x cloudera cloudera 0 B 0 0 B cloudera
drwxrwxrwx hdfs supergroup 0 B 0 0 B examples
drwxr-xr-x hdfs supergroup 0 B 0 0 B hdfs
drwxr-xr-x mapred hadoop 0 B 0 0 B history
drwxrwxrwx hive hive 0 B 0 0 B hive
drwxrwxrwx oozie oozie 0 B 0 0 B oozie
drwxr-xr-x sample sample 0 B 0 0 B sample
drwxr-xr-x spark spark 0 B 0 0 B spark


Browsing HDFS - Mozilla Firefox
Browsing HDFS
quickstart.cloudera:50070/explorer.html#/user/cloudera
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/user/cloudera
Go!
Permission Owner Group Size Replication Block Size Name
-rw-r--r-- cloudera cloudera 58 B 1 128 MB README.md
drwxr-xr-x cloudera cloudera 0 B 0 0 B demo
-rw-r--r-- cloudera cloudera 1.42 KB 1 128 MB join1_mapper.py
-rw-r--r-- cloudera cloudera 3.86 KB 1 128 MB join1_reducer.py
-rw-r--r-- cloudera cloudera 1.16 KB 1 128 MB make_join2data.py
drwxr-xr-x cloudera cloudera 0 B 0 0 B output_new
drwxr-xr-x cloudera cloudera 0 B 0 0 B output_new_2
drwxr-xr-x cloudera cloudera 0 B 0 0 B outz
drwxr-xr-x cloudera cloudera 0 B 0 0 B streaming-input
drwxr-xr-x cloudera cloudera 0 B 0 0 B streaming-output
drwxr-xr-x cloudera cloudera 0 B 0 0 B tmp
drwxr-xr-x cloudera cloudera 0 B 0 0 B wordcount


Browsing HDFS - Mozilla Firefox
Browsing HDFS
quickstart.cloudera:50070/explorer.html#/user/cloudera/tmp
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/user/cloudera/tmp
Go!
Permission
Owner
Group
Size
Replication
Block Size
Name
-rw-r--r--
cloudera
cloudera
58 B
1
128 MB
README.md
Hadoop, 2014.


Browsing HDFS - Mozilla Firefox
Browsing HDFS
quickstart.cl
Cloudera
Hue
Hadoop
Brow
/user/cloud
Go!
Permissi
-rw-r--r--
Hadoop, 20
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ more Downloads/README.md
This is a readme file.
one
two
three.
Just another file
[cloudera@quickstart ~]$
Cancel
Save File
Close

Useful references

Documentation:

- https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

Hadoop Architecture:

- http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/
- http://www.edureka.co/blog/apache-hadoop-hdfs-architecture/

Interview Questions

—*HDFS. http://www.edureka.co/blog/hadoop-interview-questions-hdfs-2/*

—*Hadoop Cluster. http://www.edureka.co/blog/hadoop-interview-questions-hadoop-cluster/*

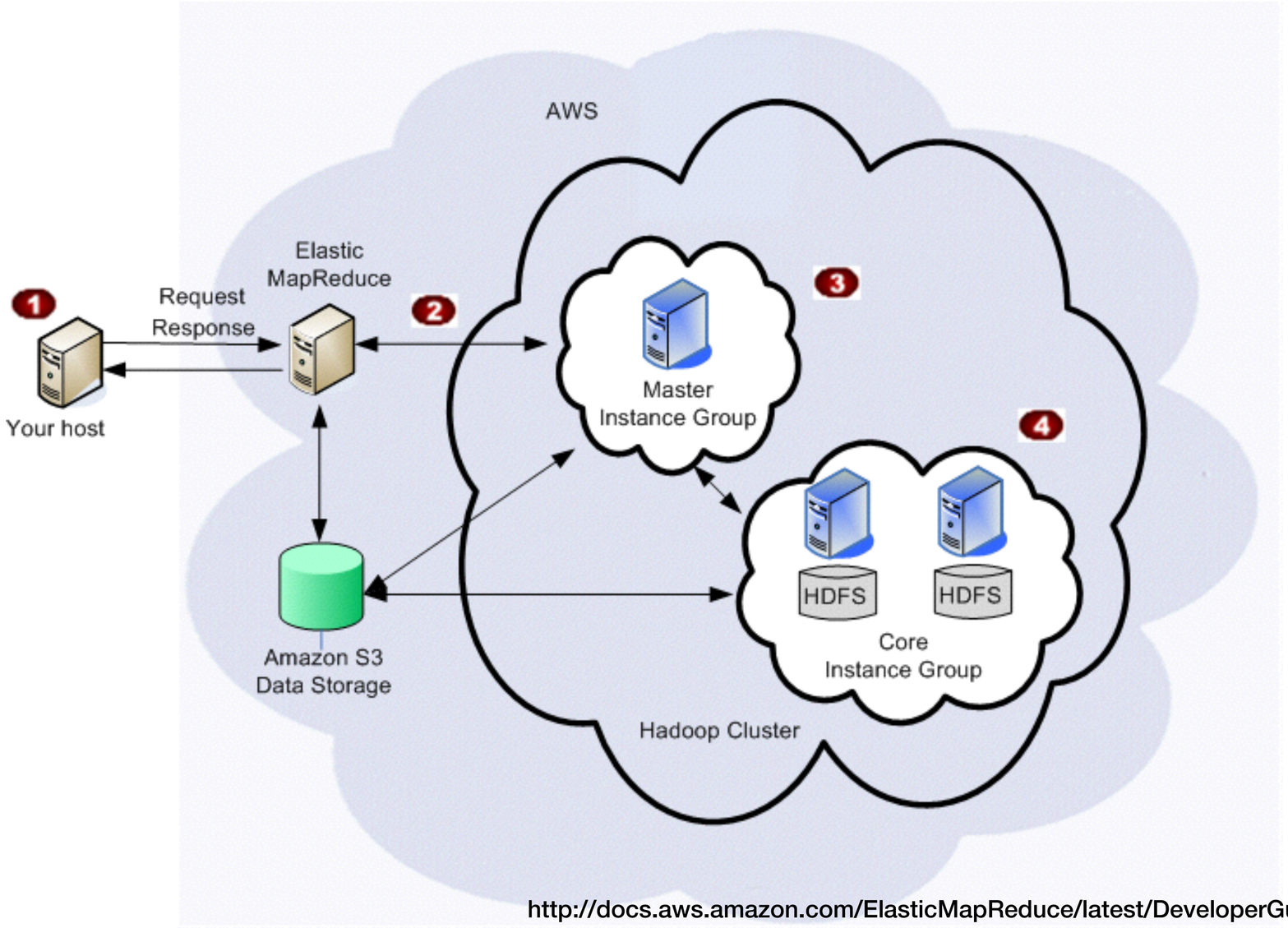
Hadoop Online zTutorial - http://hadooptutorial.info/

- Big Data | Hadoop | Map Reduce | Hive | Pig | HBase | Flume
- http://hadooptutorial.info/hadoop-certification-dump-questions/
- http://hadooptutorial.info/hadoop-interview-questions-and-answers-part-2/

Amazon Web Services

- EC2
- CloudWatch
- EBS
- EMR

Elastic Map Reduce: Amazon's Managed Hadoop Cluster



http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-manage-resize.html

AWS Educate

Amazon's "educate" program entitles you to $100 grant.

- http://aws.amazon.com/education/awseducate/
- In addition to "free tier."

Benefits to students:

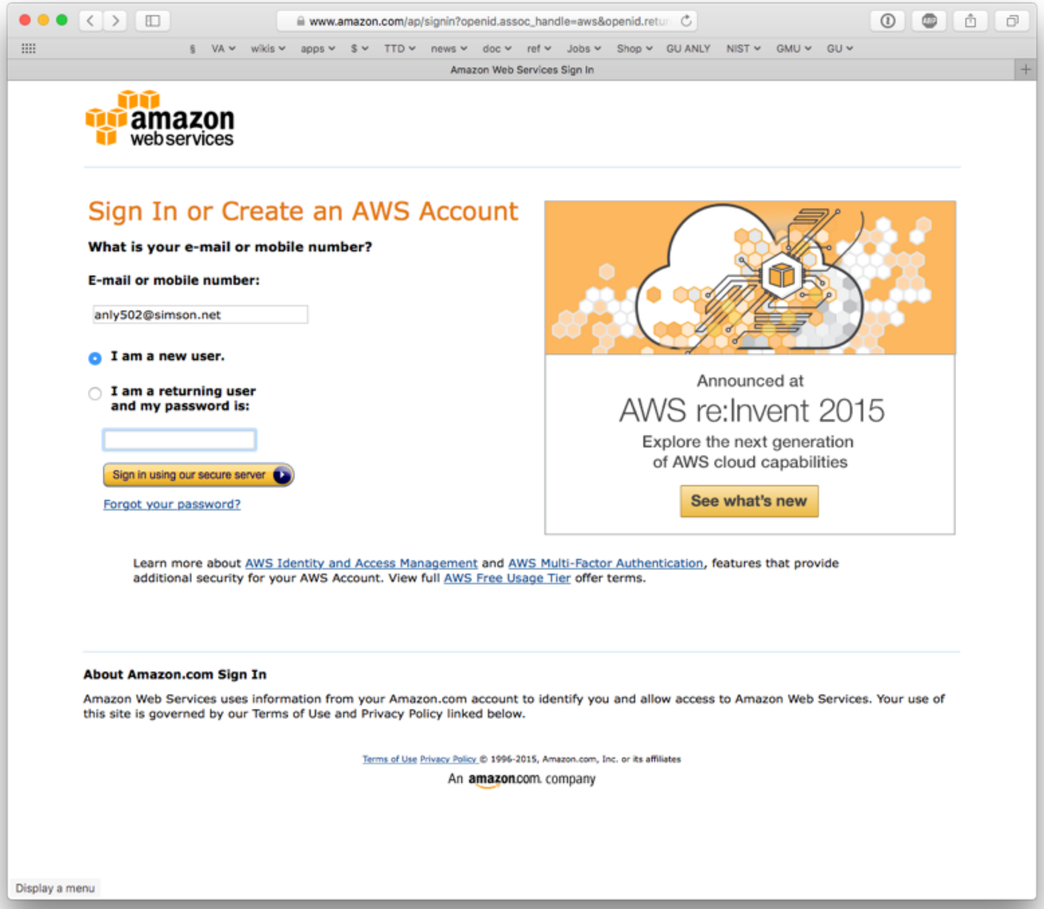
- $100/student — at member institutions (GU is a member institution)
- AWS Training — Free access to labs
- Curated Content — Free access to AWS content for homework, labs, and self-study
- Collaboration Tools — Student portal access, virtual events, provide feedback.

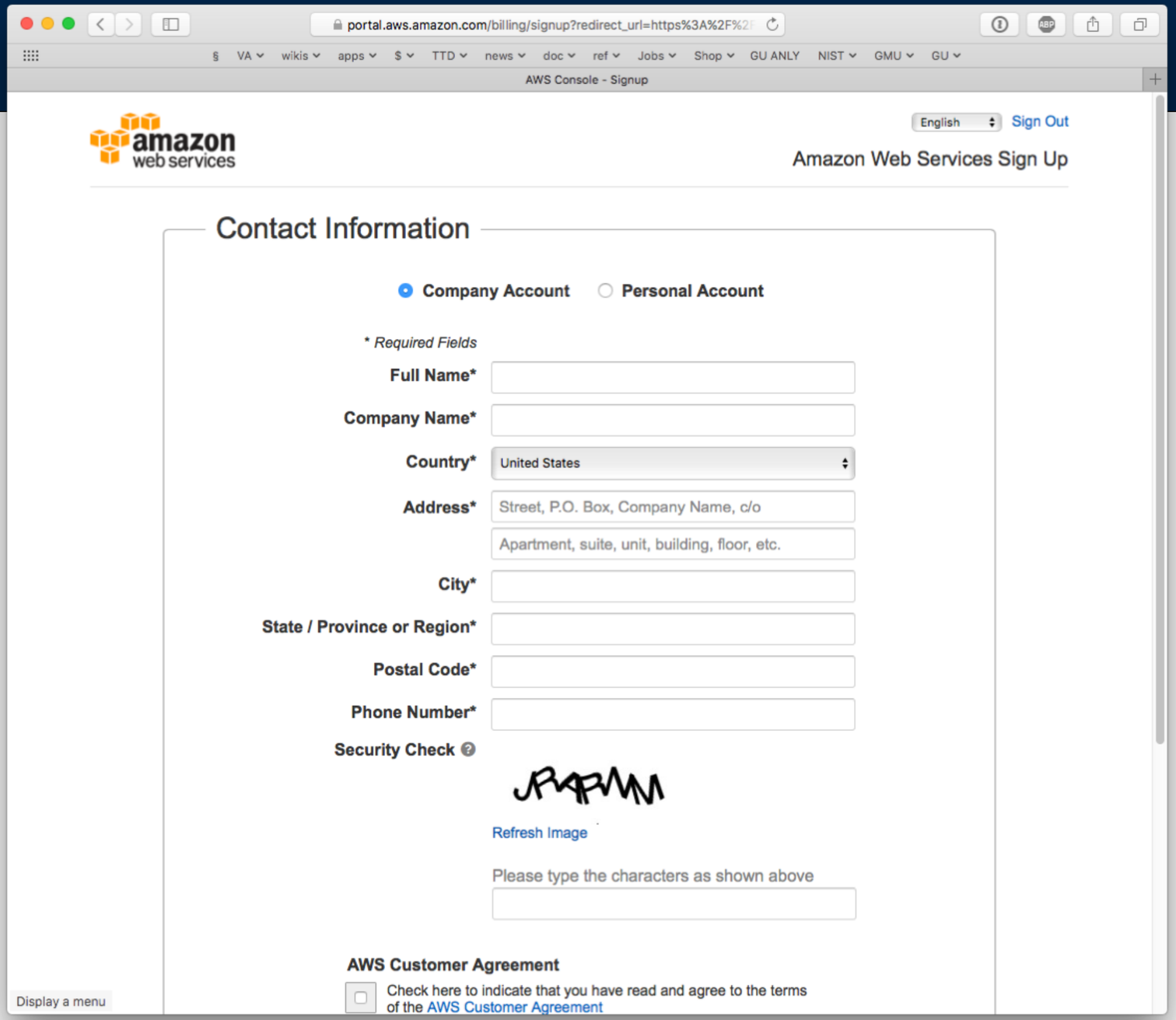
Use "free tier" to develop your code.

Use "educate" to run EMR and ES jobs.

You need an account on Amazon Web Services (AWS)

1. Create an amazon.com account (if you don't have one already)
2. Go to aws.amazon.com and sign up for Amazon Web Services

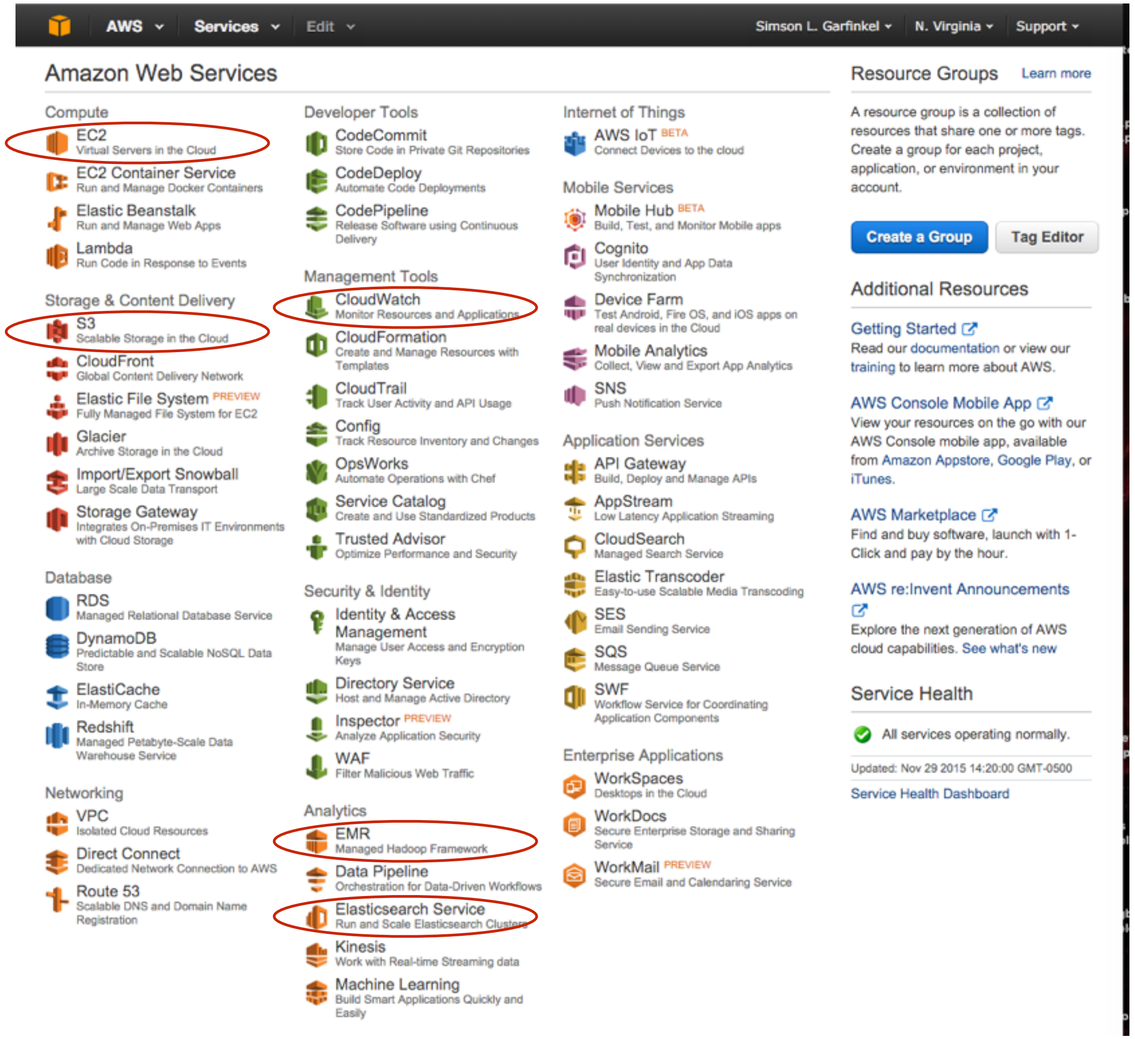



portal.aws.amazon.com/billing/signup?redirect_url=https%3A%2F%2F
AWS Console - Signup
amazon web services
English
Sign Out
Amazon Web Services Sign Up
Contact Information
Company Account
Personal Account
* Required Fields
Full Name*
Company Name*
Country*
United States
Address*
Street, P.O. Box, Company Name, c/o
Apartment, suite, unit, building, floor, etc.
City*
State / Province or Region*
Postal Code*
Phone Number*
Security Check
Refresh Image
Please type the characters as shown above
AWS Customer Agreement
Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement
Display a menu

AWS has a lot of services:

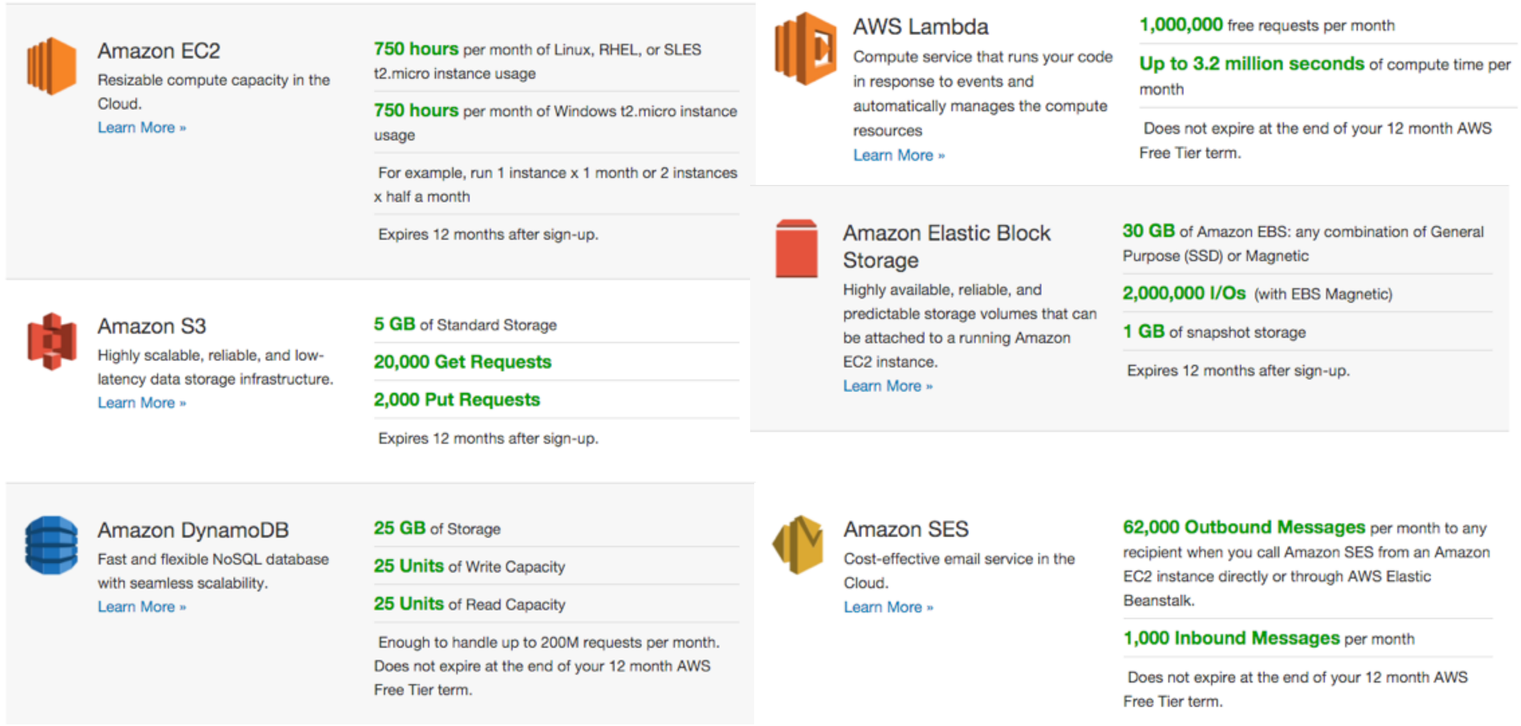
In this course, we will be focusing on:

- EC2 — Virtual Servers
- CloudWatch
- S3 — Object-based Storage
- EMR — MapReduce & Spark
- Elasticsearch — search



AWS Free Tier

- New AWS accounts are entitled to $750/month of “free tier” service.
- Includes micro instances (typically bill at $0.015 cents/hour)
- **Does not include EMR or large machines.**



Amazon EC2
Resizable compute capacity in the Cloud.
Learn More »

750 hours per month of Linux, RHEL, or SLES t2.micro instance usage

750 hours per month of Windows t2.micro instance usage

For example, run 1 instance x 1 month or 2 instances x half a month

Expires 12 months after sign-up.

Amazon S3
Highly scalable, reliable, and low-latency data storage infrastructure.
Learn More »

5 GB of Standard Storage

20,000 Get Requests

2,000 Put Requests

Expires 12 months after sign-up.

Amazon DynamoDB
Fast and flexible NoSQL database with seamless scalability.
Learn More »

25 GB of Storage

25 Units of Write Capacity

25 Units of Read Capacity

Enough to handle up to 200M requests per month. Does not expire at the end of your 12 month AWS Free Tier term.

AWS Lambda
Compute service that runs your code in response to events and automatically manages the compute resources
Learn More »

1,000,000 free requests per month

Up to 3.2 million seconds of compute time per month

Does not expire at the end of your 12 month AWS Free Tier term.

Amazon Elastic Block Storage
Highly available, reliable, and predictable storage volumes that can be attached to a running Amazon EC2 instance.
Learn More »

30 GB of Amazon EBS: any combination of General Purpose (SSD) or Magnetic

2,000,000 I/Os (with EBS Magnetic)

1 GB of snapshot storage

Expires 12 months after sign-up.

Amazon SES
Cost-effective email service in the Cloud.
Learn More »

62,000 Outbound Messages per month to any recipient when you call Amazon SES from an Amazon EC2 instance directly or through AWS Elastic Beanstalk.

1,000 Inbound Messages per month

Does not expire at the end of your 12 month AWS Free Tier term.

—*Some of these expire after 12 months, some don’t.*

- https://aws.amazon.com/free/

Amazon EC2

EC2 — Elastic Compute Cloud

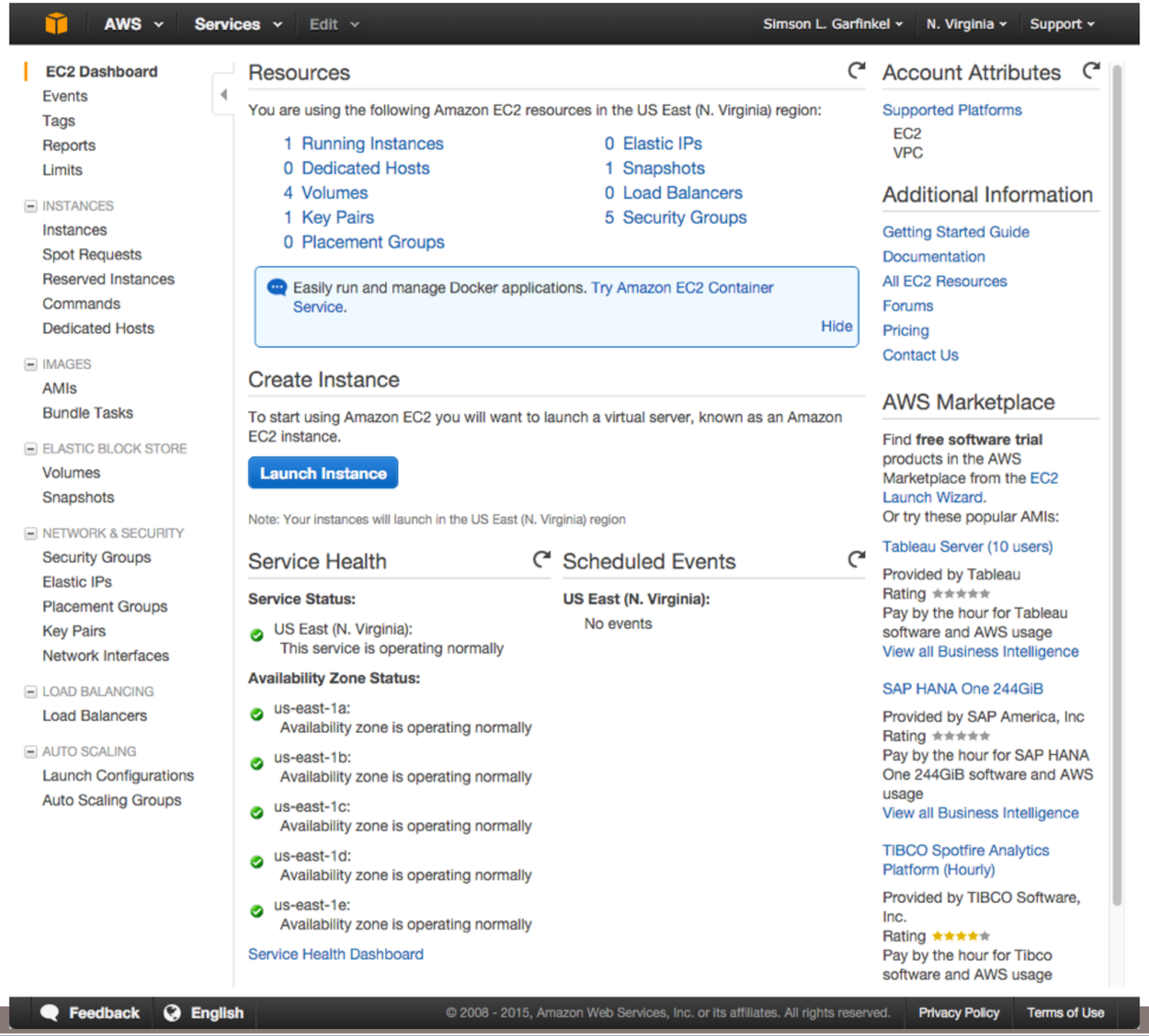
Virtual Machines in the cloud — You create “Instances”

- Horizontal Scaling — Create *many VMs.*
- Vertical Scaling — Create small and large VMs (cores, RAM, networking)
- Geographical Diversity — Create in different physical locations (“availability zones”)

Each instance has:

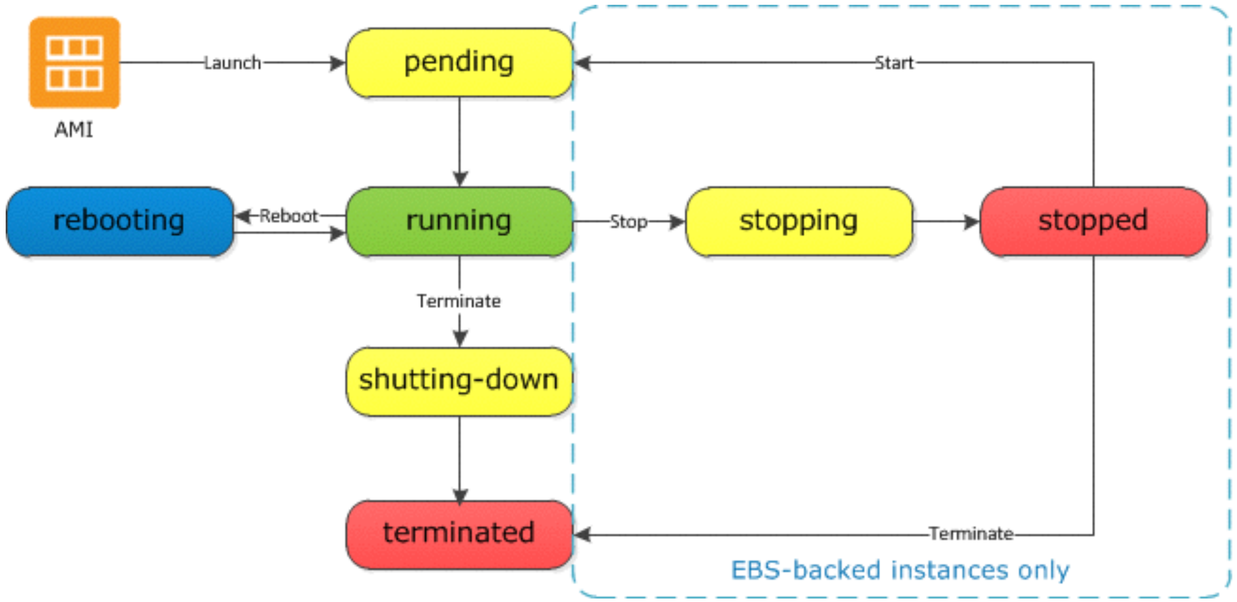
- AMI — Amazon Machine Image — the initial “boot volume”
- Virtual drives — Elastic Block Store
- Network interface and firewall

EC2 Dashboard



Instance life cycle:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

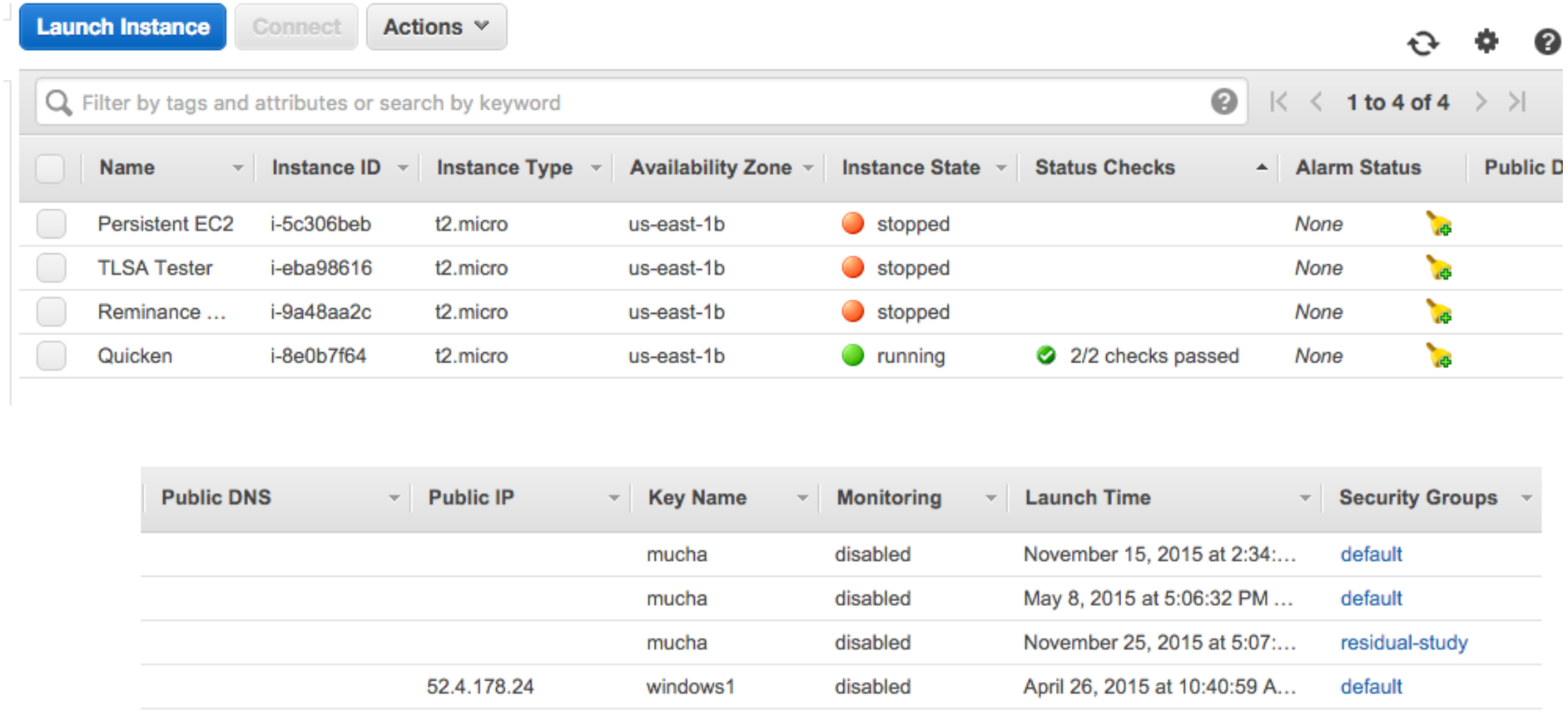


- All instances boot from an AMI (you can upload your own.)
- You specify if the EBS volume is kept or lost on termination.

You pay for:

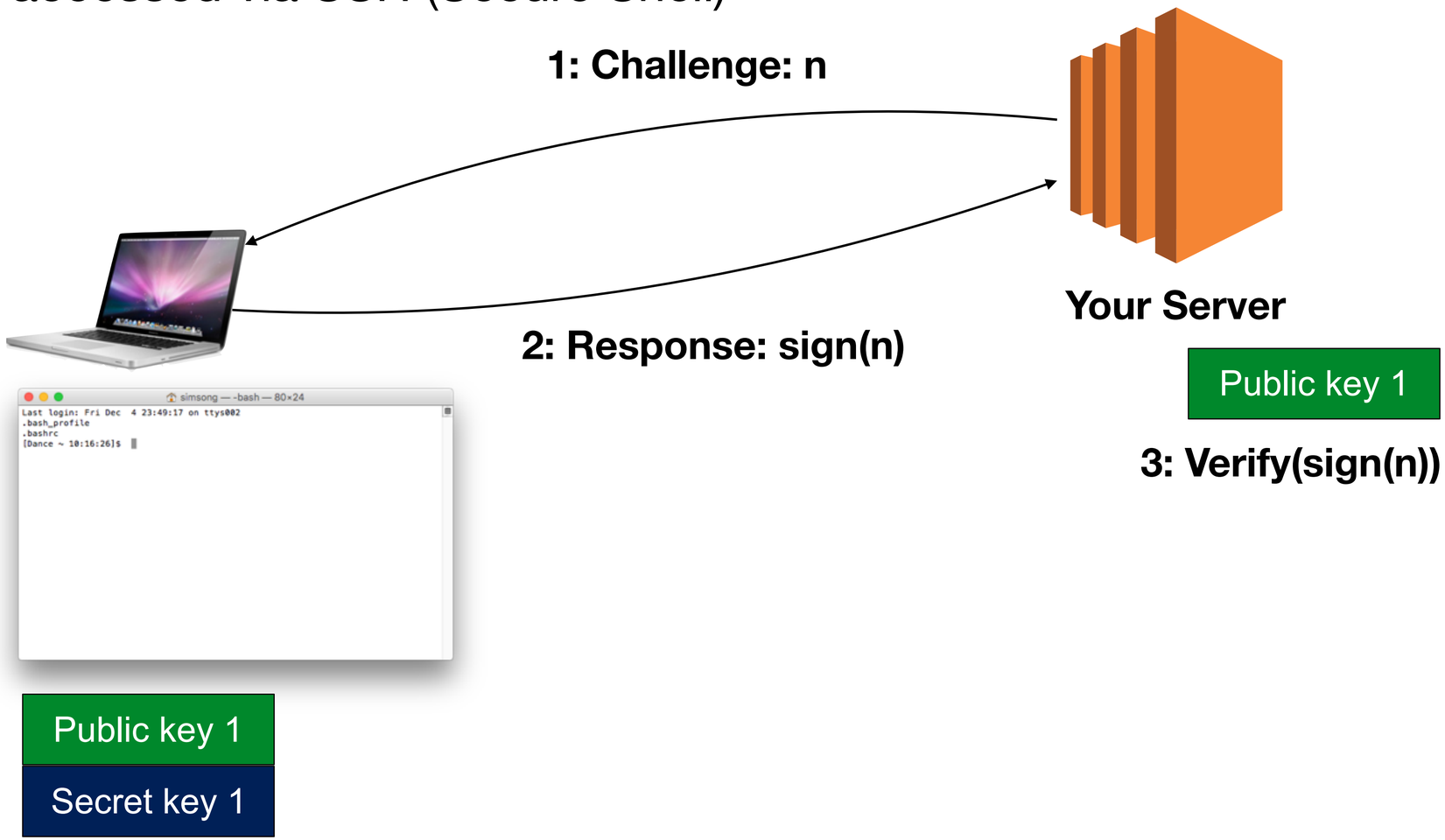
- Instances that are running
- EBS-backed storage
- Bandwidth from EC2➜Rest of Internet

Instance control panel:



Accessing your instance: AWS key pairs

Linux instances are accessed via SSH (Secure Shell)



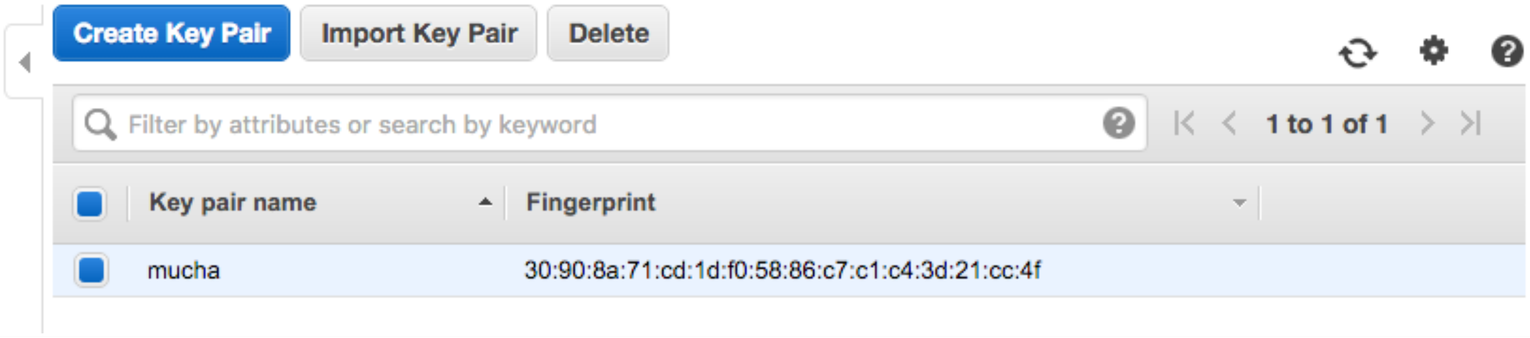
Accessing your instance: AWS key pairs

Linux instances are accessed via SSH (Secure Shell)

- AWS uses SSH “public key authentication.”
- Two ways to get your public key to Amazon:
  - *You create a public/private keypair with “ssh keygen -t rsa -f mykey.pem” & import*
  - *Amazon will create the pair and you download it.*
- You use the private key to authenticate.

Key pairs:

- Each key is identified by a “Fingerprint.”
- If you lose your private key, you can’t access your server.



Making a key…

```
$ ssh-keygen -t rsa -f mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey.
Your public key has been saved in mykey.pub.
The key fingerprint is:
SHA256:B+/MiY/KgrDy5Agc8pkP+/AKd7YbA2Gdno7PnnQmfXs simsong@Dance.local
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|   . .           |
|  o o   .        |
| . o .   o       |
|... o   S o      |
|+..* .   * .     |
|o+X.O + o =      |
|+*.%.X . +E      |
|o.=+@oo.o..      |
+----[SHA256]-----+
$ cat mykey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDKvpBsCUJBGxJiGtvQF6f3LrRWpk5EbsWCDkZ2lMAMdVg0Ee4iRUuKfo62KXge8GgOuOzSSG43L/yvn+
+LtV4s8sNYA1QtxDyZmFCYLGV0slRxkKL/xN/KwxNc5EgiP1tVNrQvAhrKUCIQDspNuDPbO5DvGxb
+YyJdUAW5X5Z3DmGaylJotM0ypMaqE5+xHQndiusg9YIy7B8xFhoKCJ5+B+HlQdiQUQULuTlD2oSxLd0Wd5MIF/OaZ+0uu9HujqDWc5TweNcHPt3ycS//
s9ITNhjoUddCd3gHCH3TH5rZwM79MpAtCZipyKvowvFjgDqvAdt6MlvULQ7wpJKT9+Tl simsong@Dance.local
$
```

The matching private key…

```
$ cat mykey
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F5A1FC20F3E4F8B7BECFE707F8972509

7Ac+l4q3dfJ+0vBLdgWx3LdxV0vAF9YuWPIZqJjm1CzjLMM1POMk9Juj4PdwvlSh
btZZ+h1zymbgcdrTd2ivALIkO0PbjlgZpnZ9Vkn4MXEjsOPq/Xa2B+tI4/i3HpUL
cCBcqwNujt6pgzoVCiv5L+33Hiiupn9d9NZbdDxWTyArzYAU7vwfb6UejGrZ8ME1
mYl3qYsBtOGhViY8AkDidTYY40t4Sz3Yk66a3ZiTcBivnOBpdIgZ1S8t3Stoaprt
jrlm5eZFDtA7hIlXdNBsgu8xmKDyGAx/bP7WFakCsoCsUQOyl53oPZcIFkivyVqV
dki8MCiGZAu0oh68L2qzzINjNfOGQCBQIosxLMWCAT5KRyGMzuxmARxgc6ZWmabJ
xwSgHT+UdNq3IEP8k/5GPxHsbmVdWe6MXmVfSziZFAPYqEYI/FqiiNVoZDMOR2pf
sSImx2DtYUhe5c4b5n3I20wEnY9gDOkzDjiW4nJ/l4Tk2HXX0yPIMeFlLbHdisU2
FPWp85x43Ibeesd/oTOwRUiPE/4a8y0QNcdsHOjefiWZBL5qgH5vr8tsgYFN5IKv
peJh2DVGQd7u1jnK1hnKIF6+TbvnQmx9RPrVX3Nab1ba9s3B539AGxIhmyoKXyEG
UwbgPYXHRmjCgf8ENulbREdvhBV99cXldk6RlsCoHZjzE1FGKiPOGpHd2fqj2PTp
ZFOXQZ2Xdnl/L26wzr5M9daltlUfa7rSllrpQFDQ2OP0LOGvLVVrmHRYmwUgv2Fb
BbxgwVtOp6MNAxNz0uqjkG+LO44GBV93eVv8aT/2s4V0/s4ul13uGDFqT+NOwYz3
/SA83Mg6u67Aoqpb0261ieeUeaalx+NECfY904tlLtnnYsKnbx6FiThgQXTxRf7R
iXpUQNCkNG0CAaKb4jdlNj2vb99VRfW42Ldoh6pGWCCXAXRmJ5O18s+bqySfSYGP
+n+f45+yPZRbnxGujZRWOZ/apmCcVyNIBsv3+3smW6ISz7jtXZPFxxRcrOnUr2VT
YnUgDJWskwB+aJpPn8KbvOijjlTOi5k3KgdOjVgTvzLhuOsCPrxPnuET8LG/e5PZ
sq3vsn2hWhcqKDSmzyXL0iFkAqByOxIh5hLMQJ5yGz3RnH9yJEBv8xpXFISjxgmg
op2HwN2e08HuZZQ2gRAIdZgTOJVD+hTv+fbBpV+sTGVLqcVxxjHAOX1WvZgeB2Ax
EH0fXBjthyas9GBjJ3EQtHpcFKQpj+HPX0IdkpYn35BSED3I9mn1OeWcLugvMb9l
OUHTcyw5Gi2sAxdrxNTt9XXFsNiSdxkdPlQHqJe94KilFrPWgYBVj8c39fynI8qL
/N0NOZS/S/FCdua05wIF40LuZzTqtkB5A7CYZinRQiBiTGNJAq0uB7wkU6gu+woj
t33rN6cuoim/SNxQiyJhSgHLF4nRMY+z6Yly7x6sZCBgUJcqvxFyJlhFc13fL4JE
tXAIhyiDV0e8fkr7+yGw6firlUuV1X+eZG4SDAD109phhsdRKjIEw+QBfpE8o7B1
vkRdkjAofY5rm3kzjxnInjbT1FXnwo9r6iIbJ3v0ExLRTmjga9UhNdz3qtuc6Bkx
-----END RSA PRIVATE KEY-----
$
```

Private keys can be encrypted or decrypted.

You should always store your keys encrypted.

AWS requires decrypted keys for upload.

Sncrypt & decrypt with ssh-keygen

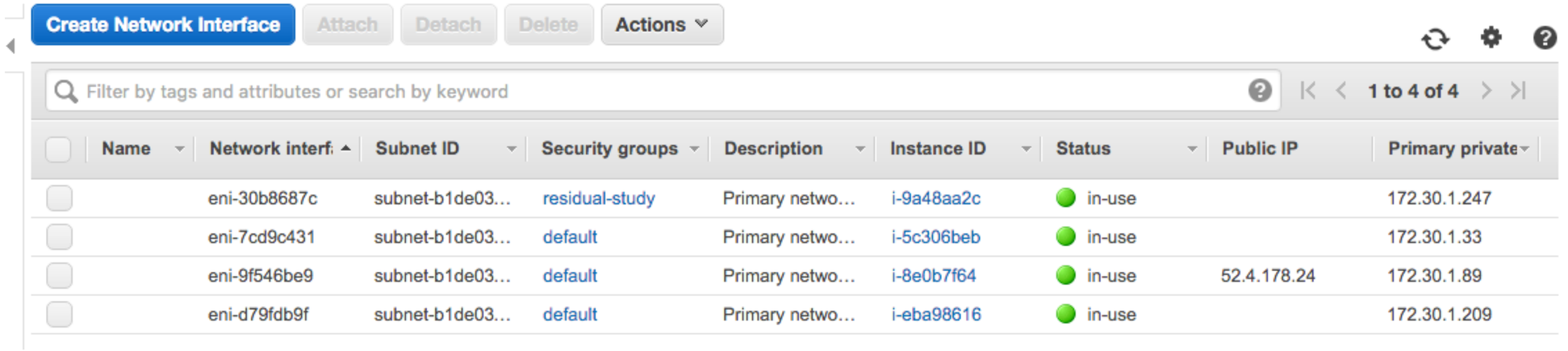
Network interfaces: each instance has 1 “virtual” interface, but possibly 2 IP addresses.

Amazon assigns a private IP address and a public IP address.

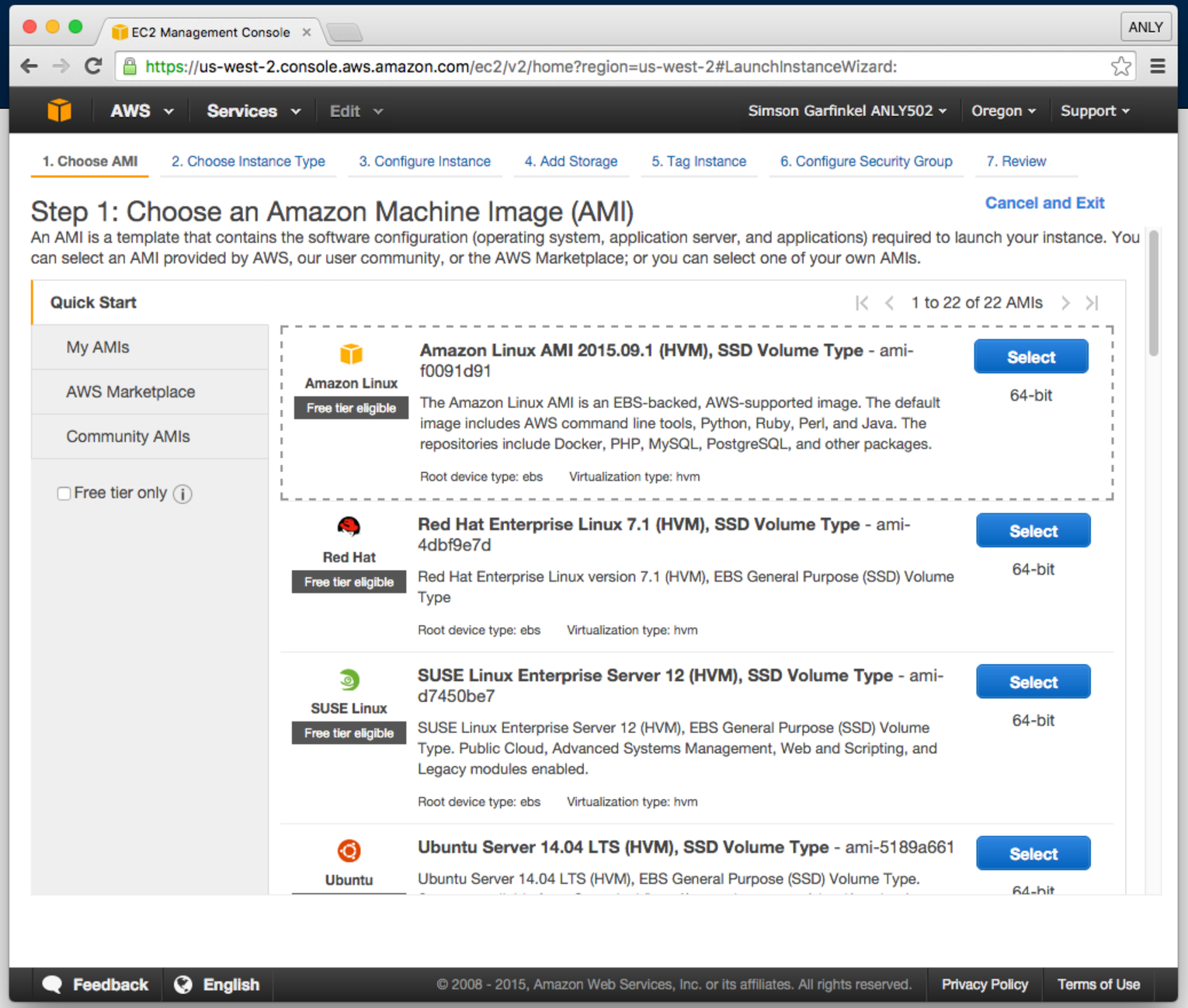
- Private IP address is the “real” address on your private subnet.
- Amazon uses two-way NAT to provide the “public” address.
- NAT implements firewall through “security groups.”

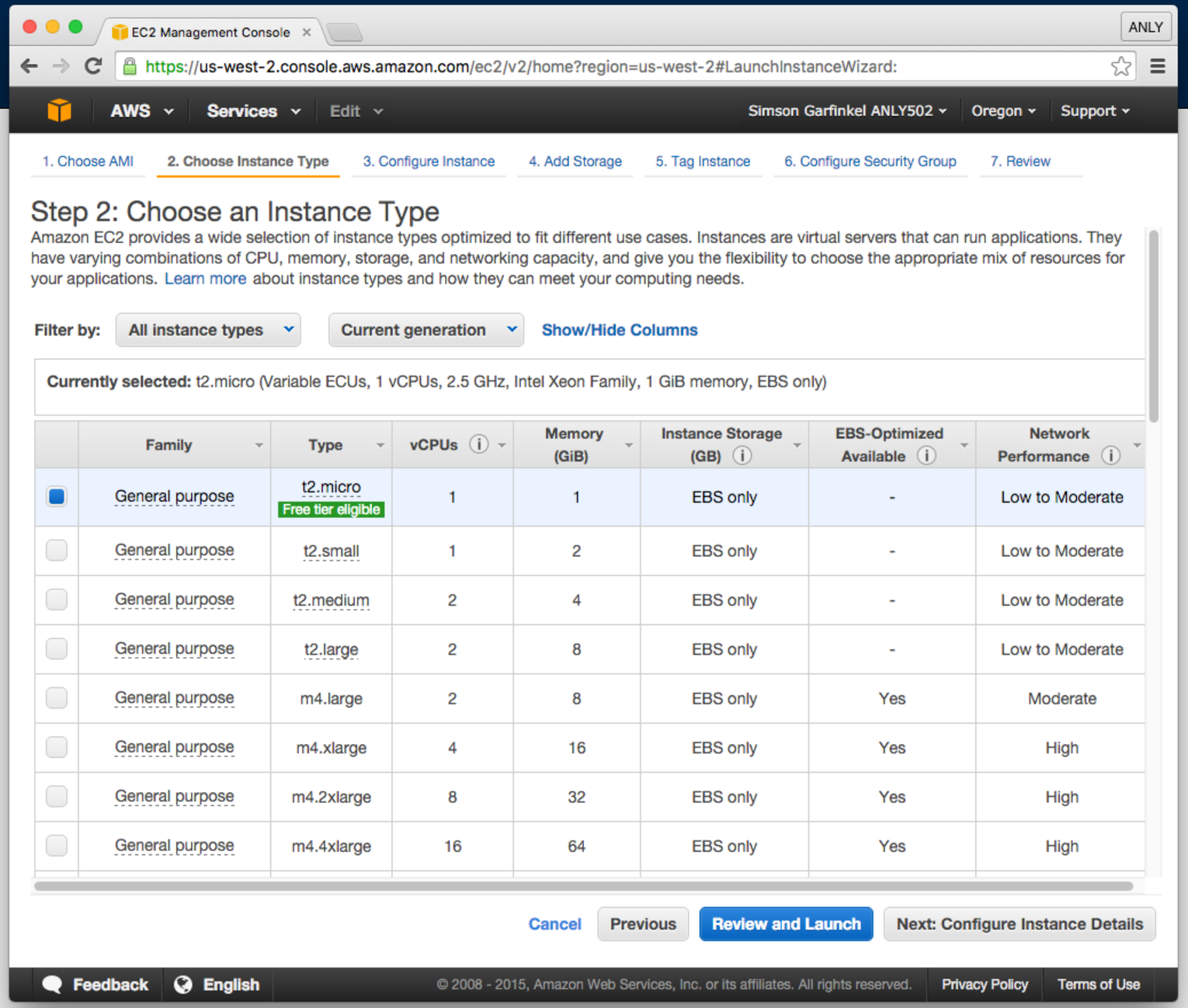
Other options:

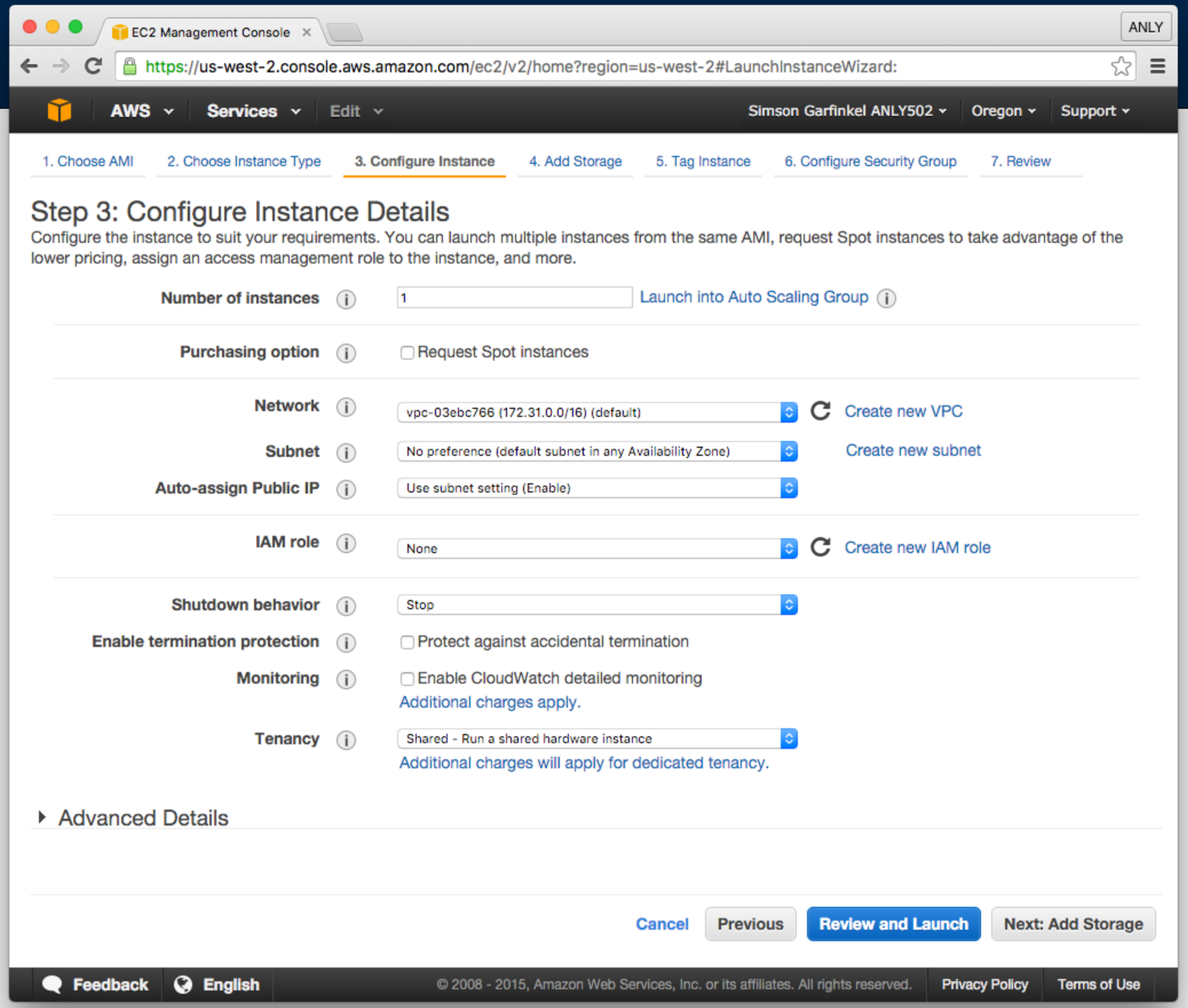
- You can have only private addresses. (More secure.)
- VPN to your organization. (Not in this course.)



Putting it all together…




EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.
Filter by:
All instance types
Current generation
Show/Hide Columns
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)
Family
Type
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
General purpose
t2.micro
Free tier eligible
1
1
EBS only
-
Low to Moderate
General purpose
t2.small
1
2
EBS only
-
Low to Moderate
General purpose
t2.medium
2
4
EBS only
-
Low to Moderate
General purpose
t2.large
2
8
EBS only
-
Low to Moderate
General purpose
m4.large
2
8
EBS only
Yes
Moderate
General purpose
m4.xlarge
4
16
EBS only
Yes
High
General purpose
m4.2xlarge
8
32
EBS only
Yes
High
General purpose
m4.4xlarge
16
64
EBS only
Yes
High
Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances
1
Launch into Auto Scaling Group
Purchasing option
Request Spot instances
Network
vpc-03ebc766 (172.31.0.0/16) (default)
Create new VPC
Subnet
No preference (default subnet in any Availability Zone)
Create new subnet
Auto-assign Public IP
Use subnet setting (Enable)
IAM role
None
Create new IAM role
Shutdown behavior
Stop
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Additional charges apply.
Tenancy
Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.
Advanced Details
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Virtualization types

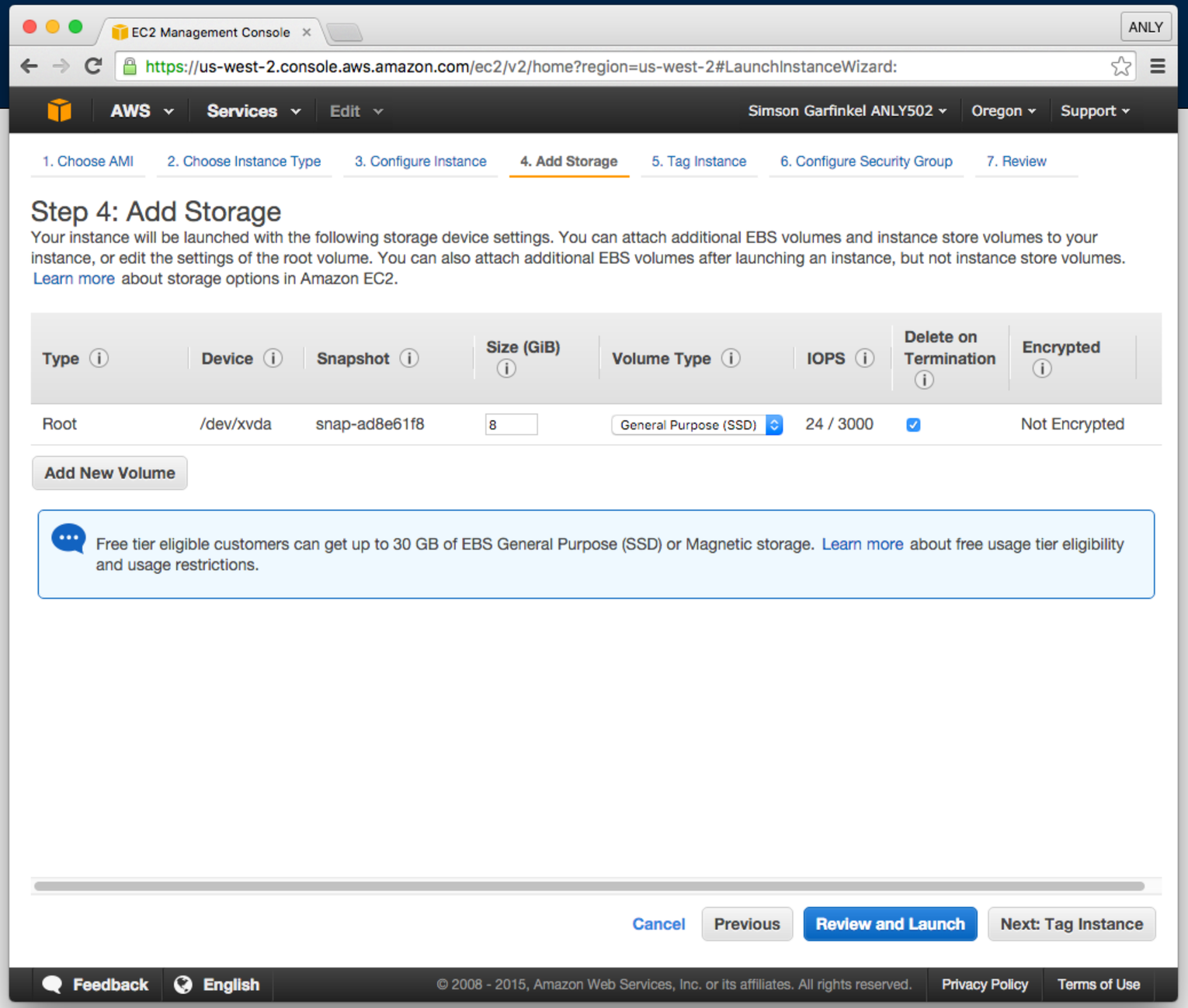
Hardware Virtual Machine (HMV)— uses hardware support

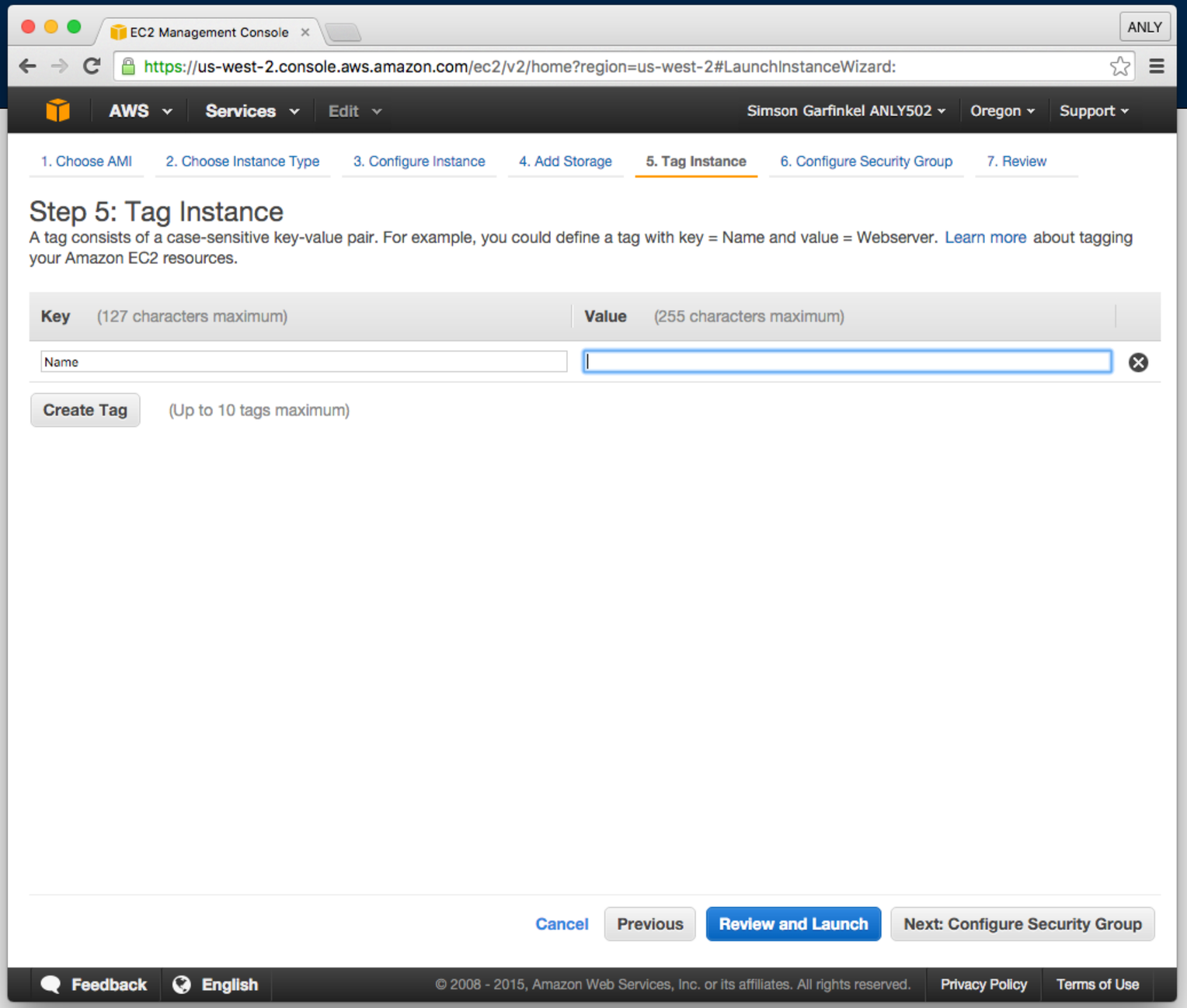
- Operating system boots normally; works with any OS (including Windows)
- Hardware extensions provide direct contact with host operating system.
- Can access GPUs, enhanced networking, etc.
- Recommended

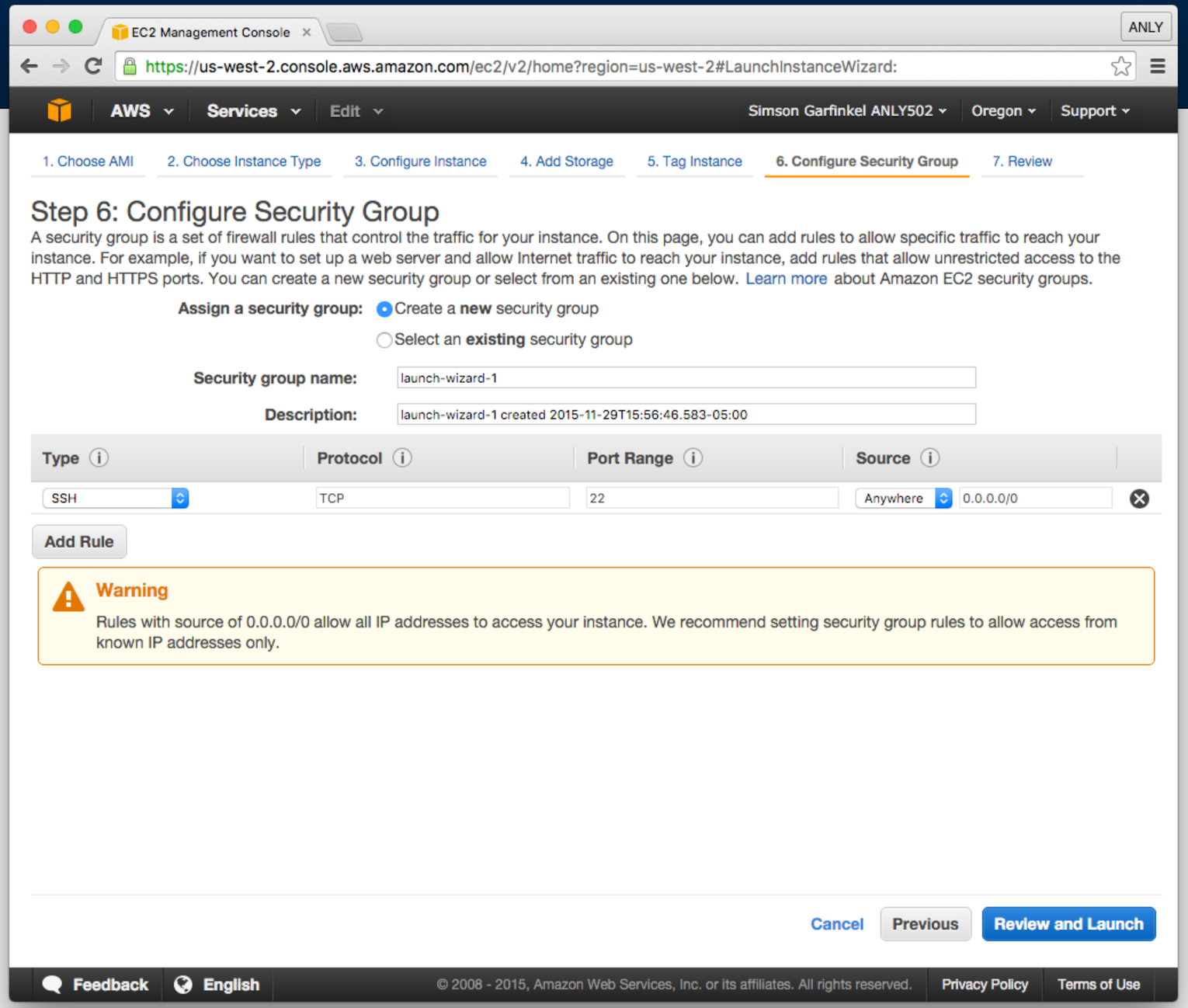
Paravirtualization (PV) — Uses software virtualization

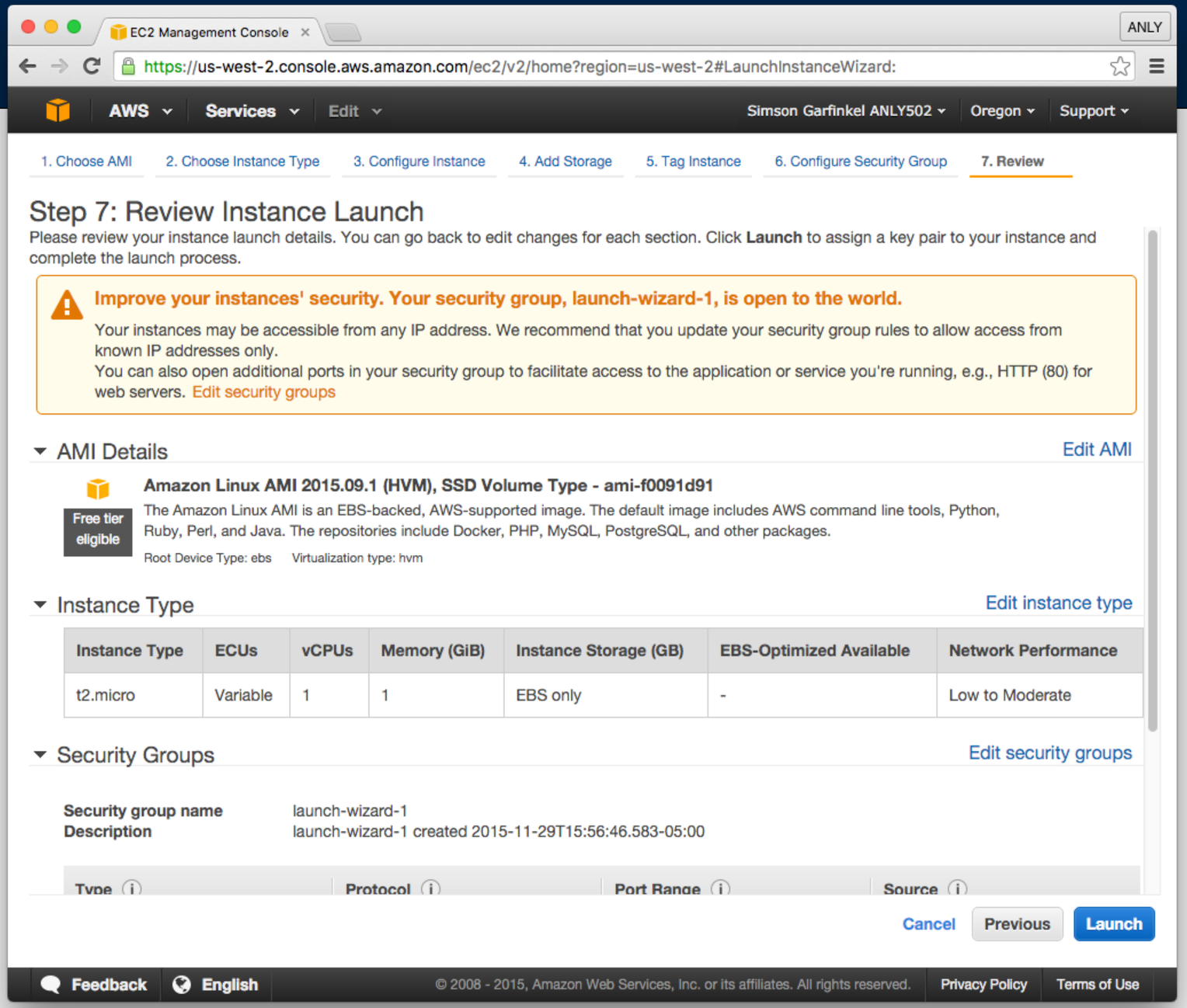
- Can run on legacy hardware (or if HVM is disabled in BIOS)
- Requires support is guest operating system. (Won't work with Windows.)

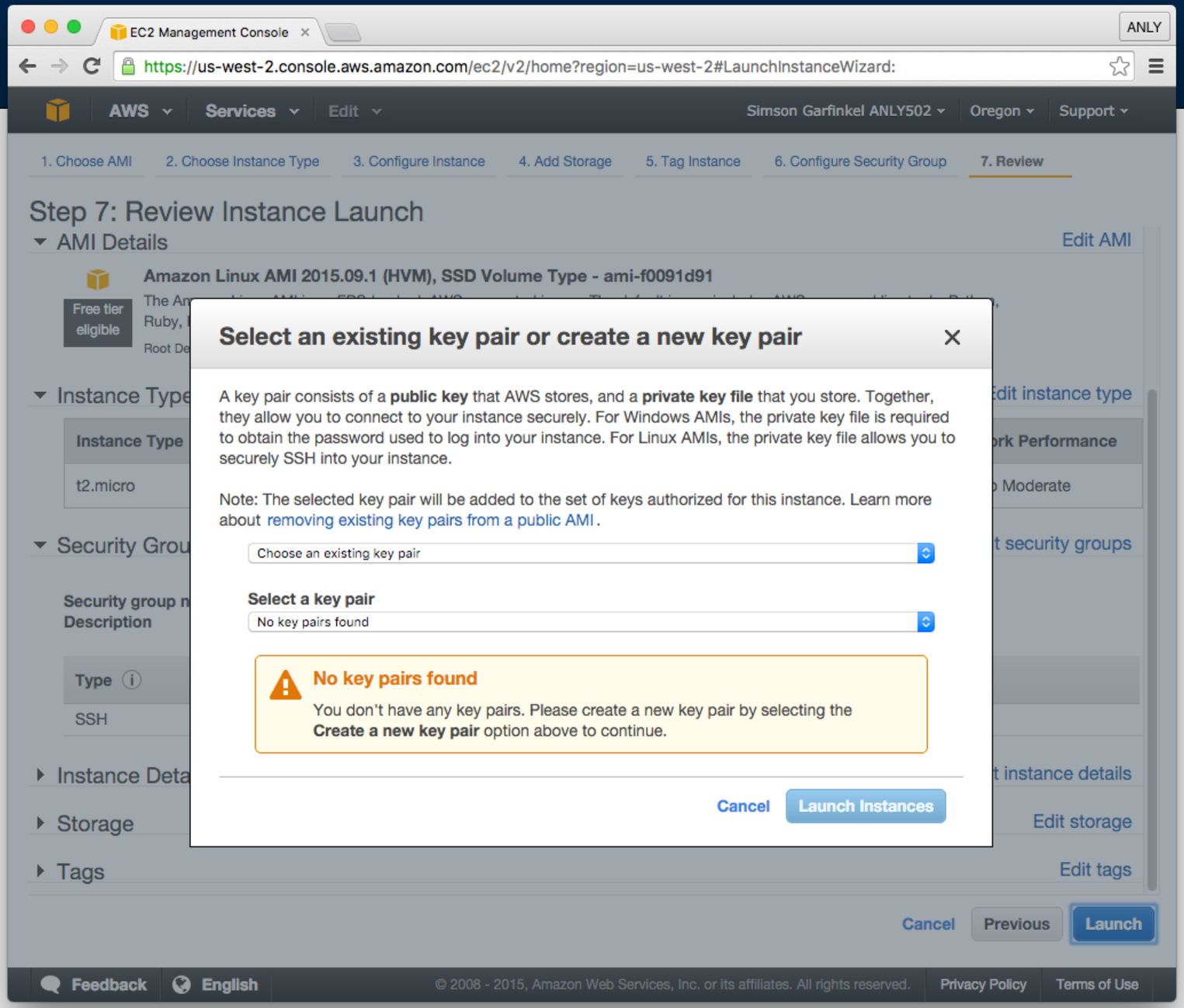
—http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html

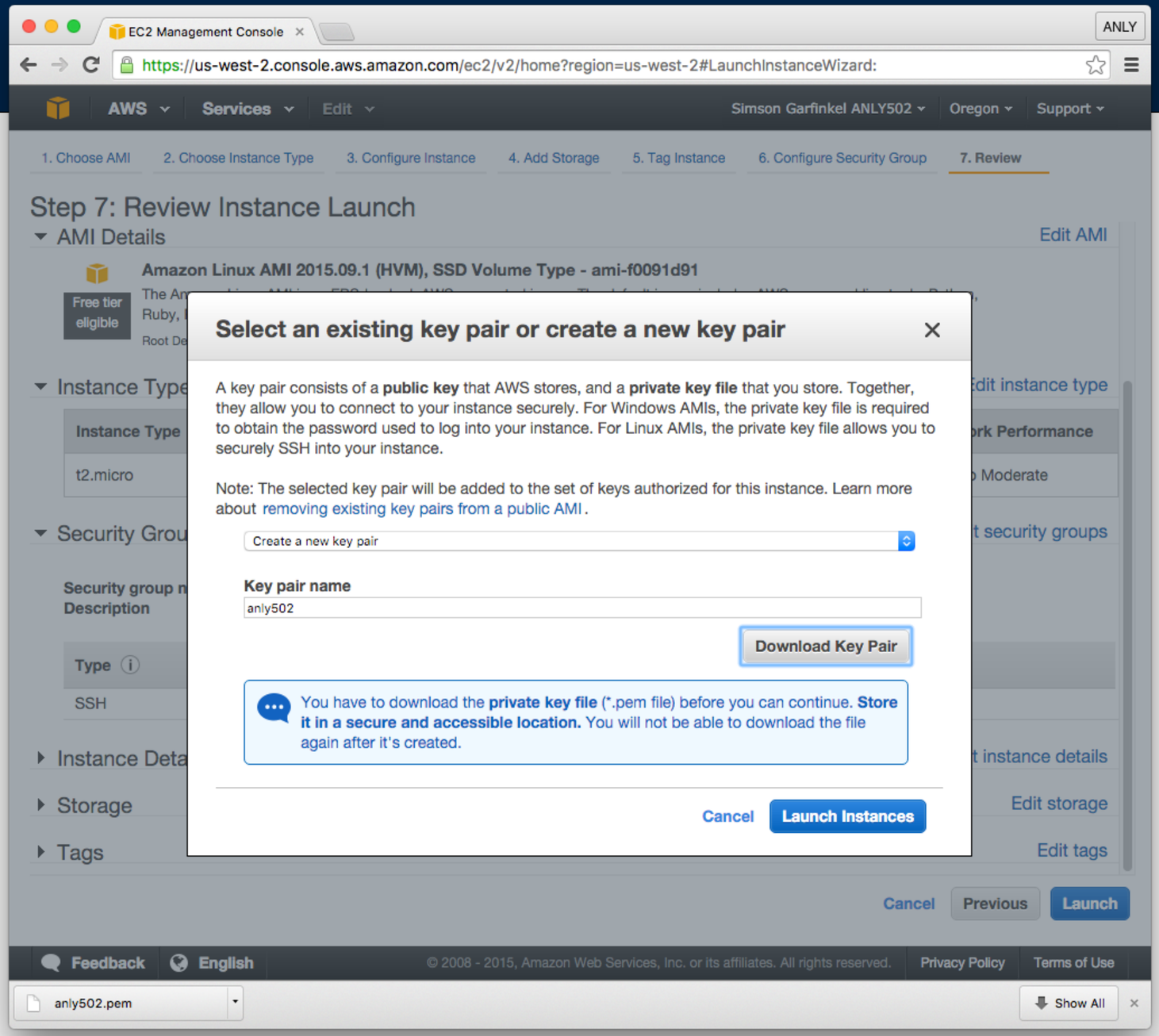

EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes.
Learn more about storage options in Amazon EC2.
Type
Device
Snapshot
Size (GiB)
Volume Type
IOPS
Delete on Termination
Encrypted
Root
/dev/xvda
snap-ad8e61f8
8
General Purpose (SSD)
24 / 3000
Not Encrypted
Add New Volume
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.
Cancel
Previous
Review and Launch
Next: Tag Instance
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

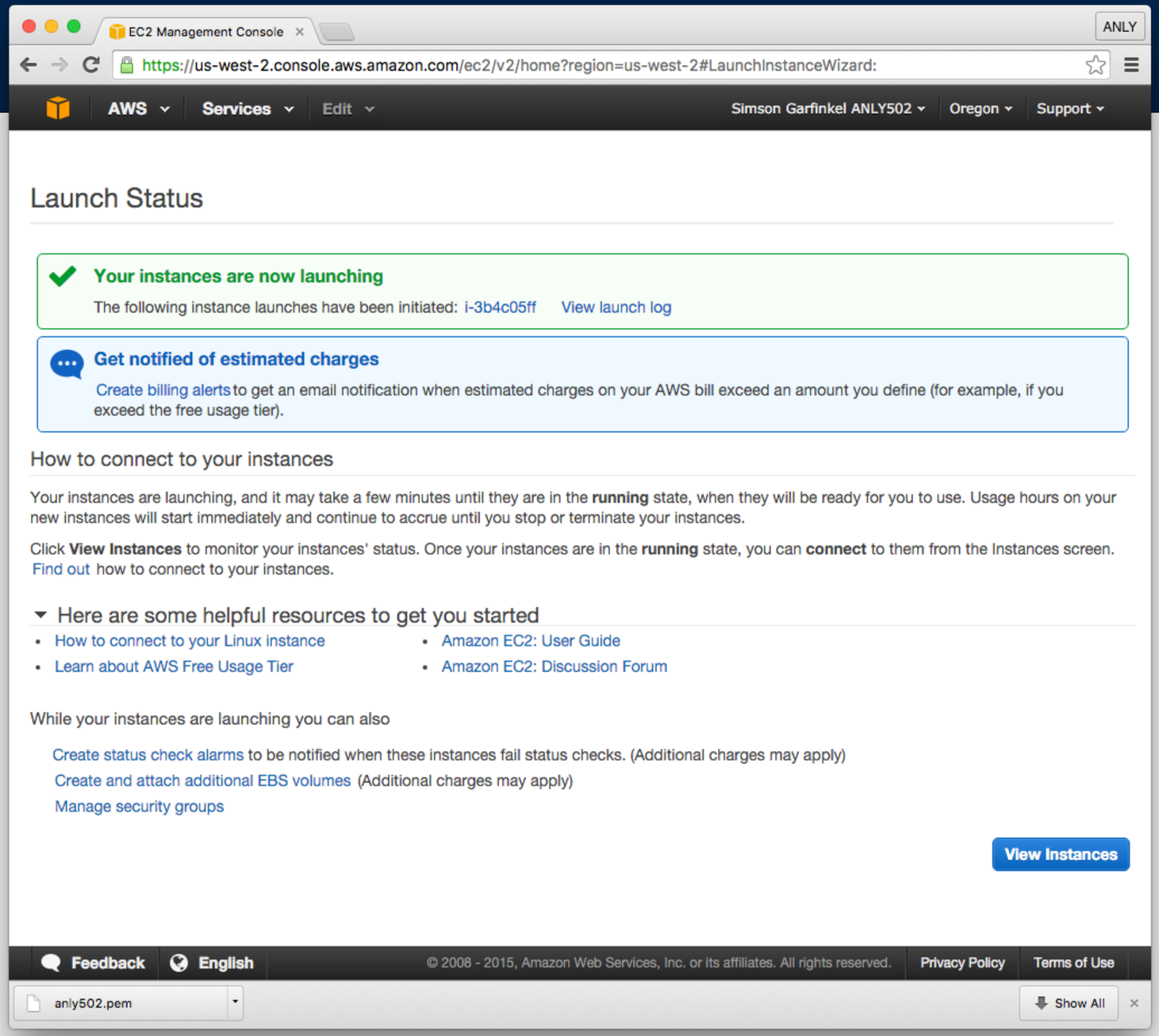

EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 5: Tag Instance
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.
Key (127 characters maximum)
Value (255 characters maximum)
Name
Create Tag
(Up to 10 tags maximum)
Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

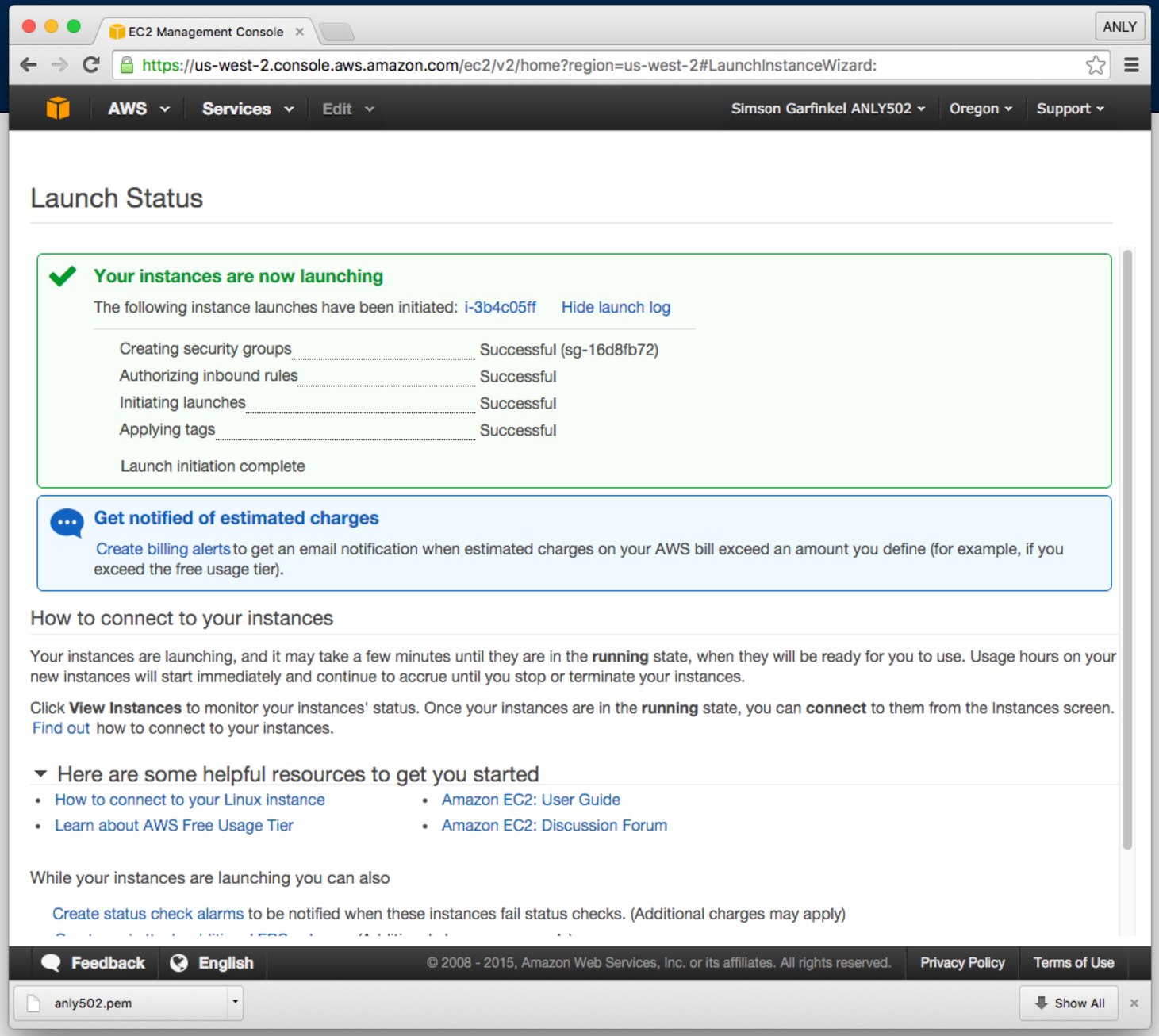

EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group:
Create a new security group
Select an existing security group
Security group name:
launch-wizard-1
Description:
launch-wizard-1 created 2015-11-29T15:56:46.583-05:00
Type
Protocol
Port Range
Source
SSH
TCP
22
Anywhere
0.0.0.0/0
Add Rule
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Previous
Review and Launch
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
Improve your instances' security. Your security group, launch-wizard-1, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups
AMI Details
Edit AMI
Free tier eligible
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance
t2.micro | Variable | 1 | 1 | EBS only | - | Low to Moderate
Security Groups
Edit security groups
Security group name launch-wizard-1
Description launch-wizard-1 created 2015-11-29T15:56:46.583-05:00
Type
Protocol
Port Range
Source
Cancel
Previous
Launch
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

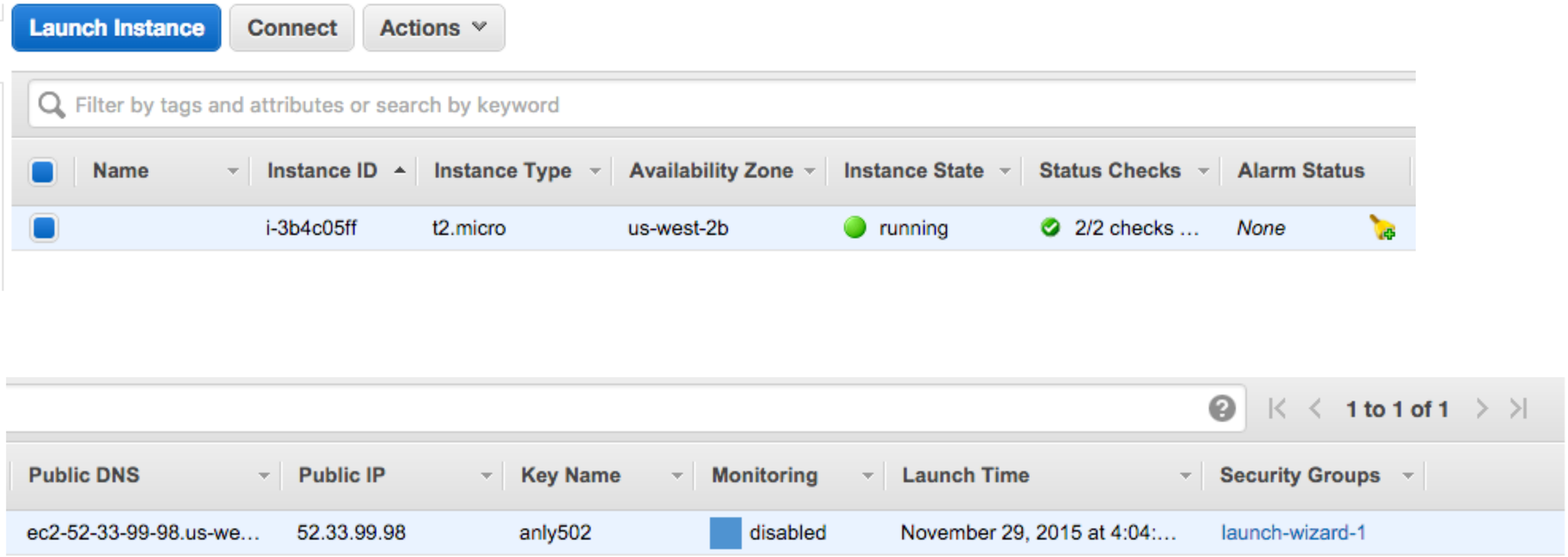

EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
AMI Details
Edit AMI
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91
Free tier eligible
Instance Type
Edit instance type
Instance Type
t2.micro
Moderate
Security Group
Edit security groups
Type
SSH
Instance Details
Edit instance details
Storage
Edit storage
Tags
Edit tags
Cancel
Previous
Launch
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Choose an existing key pair
Select a key pair
No key pairs found
No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the Create a new key pair option above to continue.
Cancel
Launch Instances
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
AMI Details
Edit AMI
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91
Free tier eligible
Instance Type
Edit instance type
Instance Type
t2.micro
Moderate
Security Group
Security group name
Description
Type
SSH
Instance Details
Storage
Edit storage
Tags
Edit tags
Cancel
Previous
Launch
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Create a new key pair
Key pair name
anly502
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.
Cancel
Launch Instances
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
anly502.pem
Show All


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
Launch Status
Your instances are now launching
The following instance launches have been initiated: i-3b4c05ff View launch log
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click View Instances to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. Find out how to connect to your instances.
Here are some helpful resources to get you started
How to connect to your Linux instance
Learn about AWS Free Usage Tier
Amazon EC2: User Guide
Amazon EC2: Discussion Forum
While your instances are launching you can also
Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
Create and attach additional EBS volumes (Additional charges may apply)
Manage security groups
View Instances
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
anly502.pem
Show All


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support
Launch Status
Your instances are now launching
The following instance launches have been initiated: i-3b4c05ff
Hide launch log
Creating security groups ..... Successful (sg-16d8fb72)
Authorizing inbound rules ..... Successful
Initiating launches ..... Successful
Applying tags ..... Successful
Launch initiation complete
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click View Instances to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen.
Find out how to connect to your instances.
Here are some helpful resources to get you started
How to connect to your Linux instance
Learn about AWS Free Usage Tier
Amazon EC2: User Guide
Amazon EC2: Discussion Forum
While your instances are launching you can also
Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
anly502.pem
Show All

Instance is running…



Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Move the private key into place:

```
$ chmod 600 Downloads/anly502.pem
$ mv Downloads/anly502.pem  ~/.ssh/
```

Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Move the private key into place:

```
$ chmod 600 Downloads/anly502.pem
$ mv Downloads/anly502.pem  ~/.ssh/
```

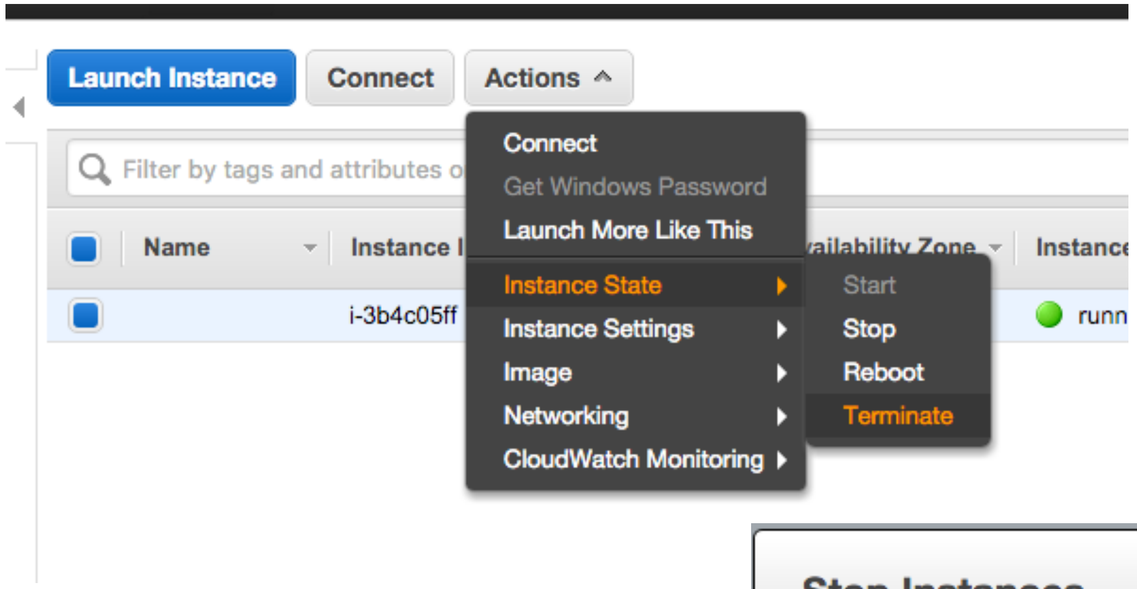
And connect!

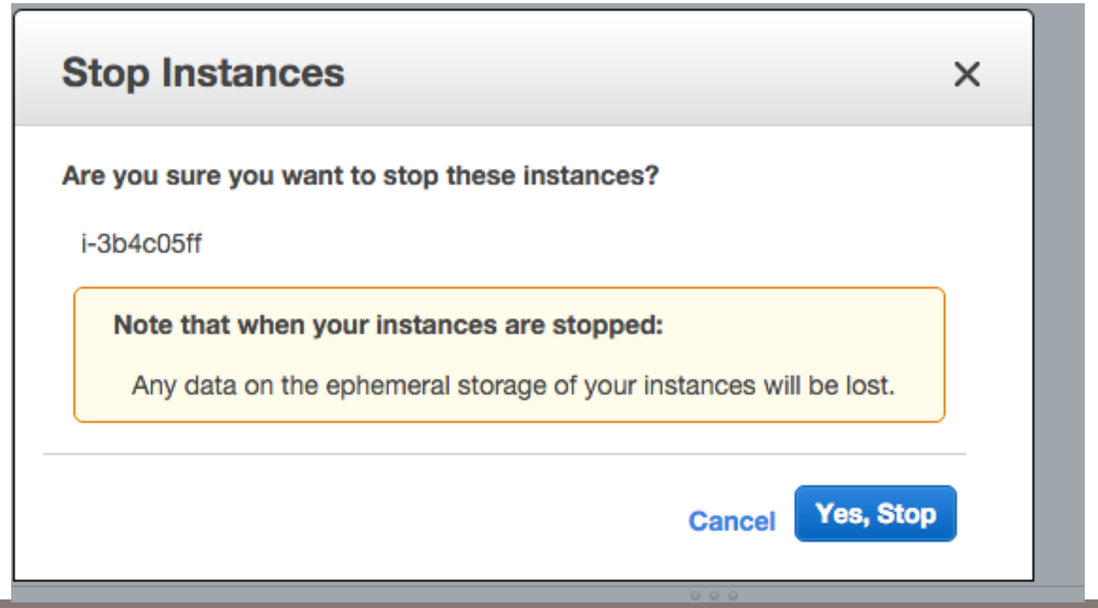
```
$ ssh -i ~/.ssh/anly502.pem ec2-user@52.33.99.98

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
3 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
$
```

We have a running instance!

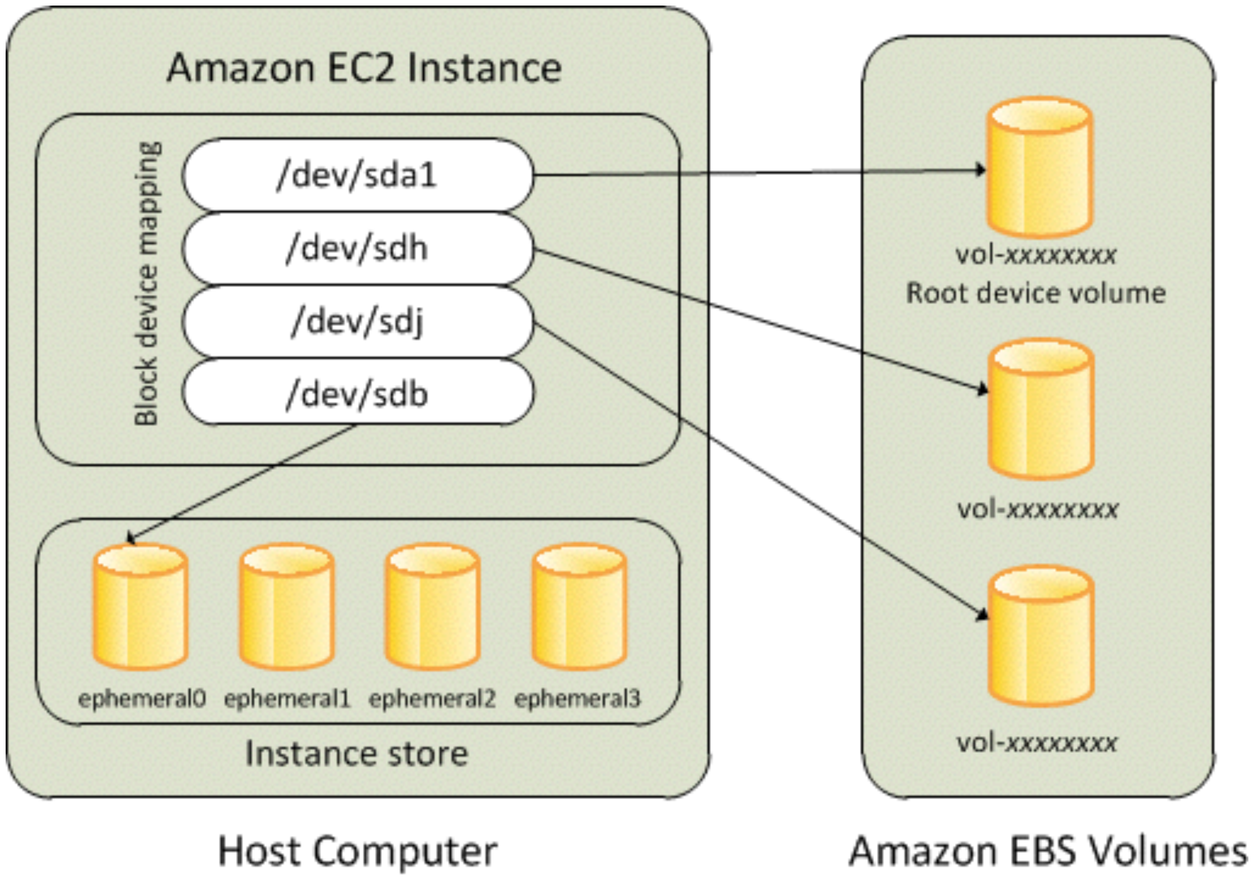
```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.1G  6.6G  15% /
devtmpfs        489M   56K  489M   1% /dev
tmpfs           498M     0  498M   0% /dev/shm
$
$ top
top - 21:46:19 up 40 min,  1 user,  load average: 0.03, 0.04, 0.03
Tasks:  68 total,   1 running,  67 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1019452k total,   326648k used,   692804k free,     9572k buffers
Swap:        0k total,        0k used,        0k free,   265376k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      20   0 19612 2536 2216 S  0.0  0.2   0:00.79 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.01 ksoftirqd/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.09 kworker/0:0
    5 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.01 kworker/u30:0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.03 rcu_sched
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
   10 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kdevtmpfs
   12 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   13 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 perf
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u30:1
   16 root      20   0     0    0    0 S  0.0  0.0   0:00.01 xenwatch
   21 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenbus
```

Don't forget to shut down when done!



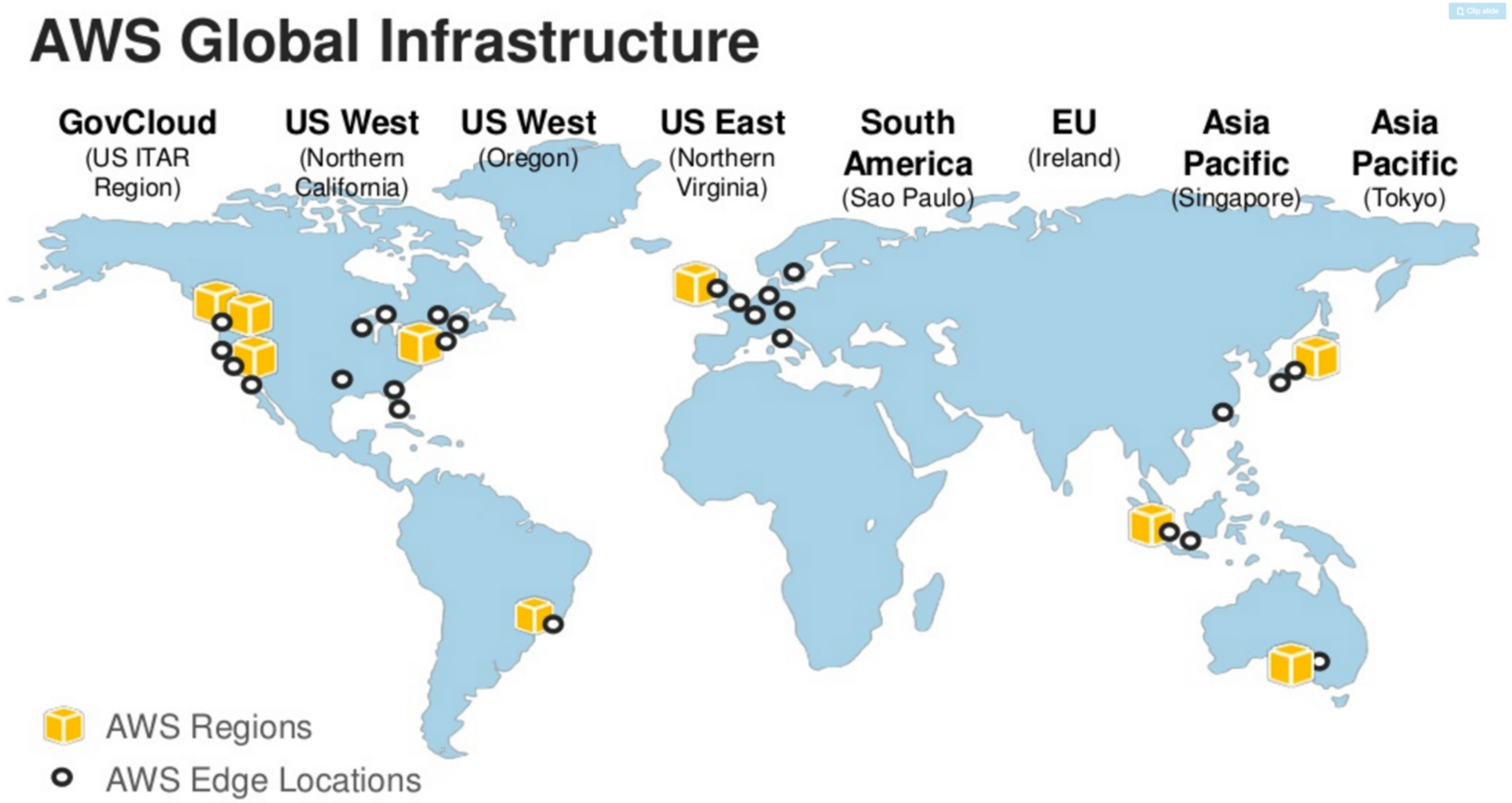


Ephemeral storage — part of the instance (local drives) faster.
EBS — separate devices — slower, but can persist.



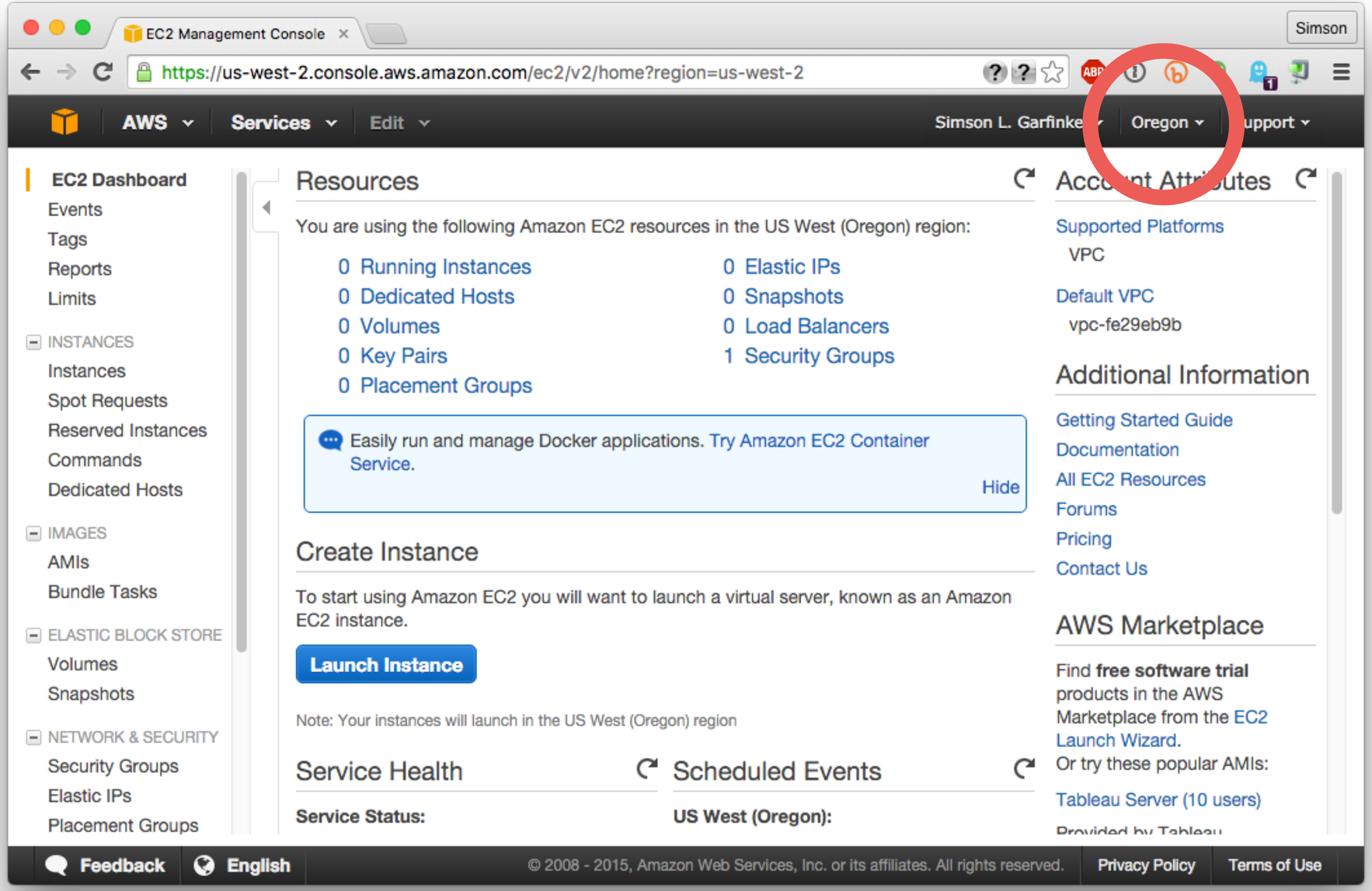
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/block-device-mapping-concepts.html

AWS has multiple regions.
They are largely independent.

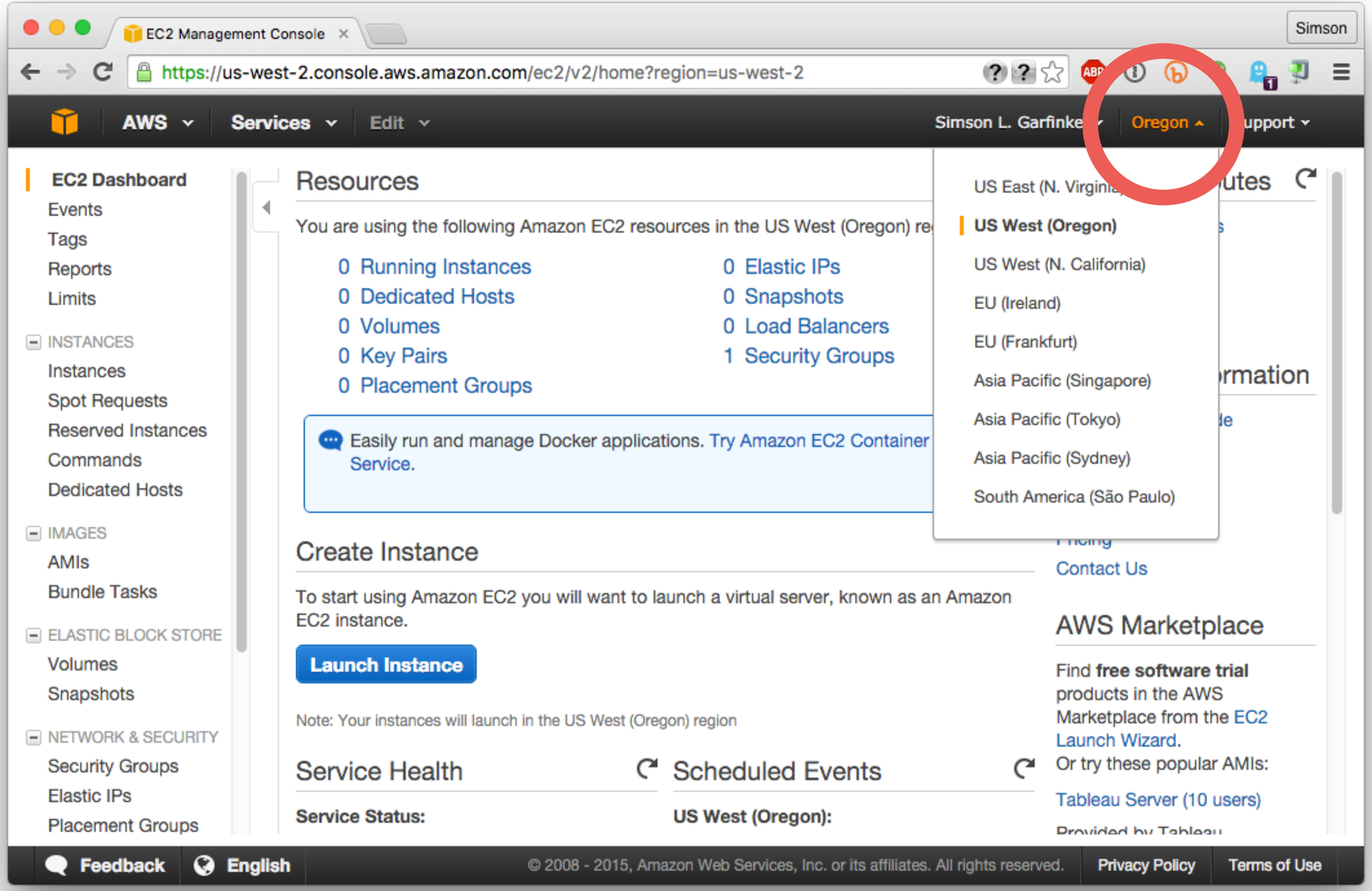


http://www.slideshare.net/my2108/what-is-aws

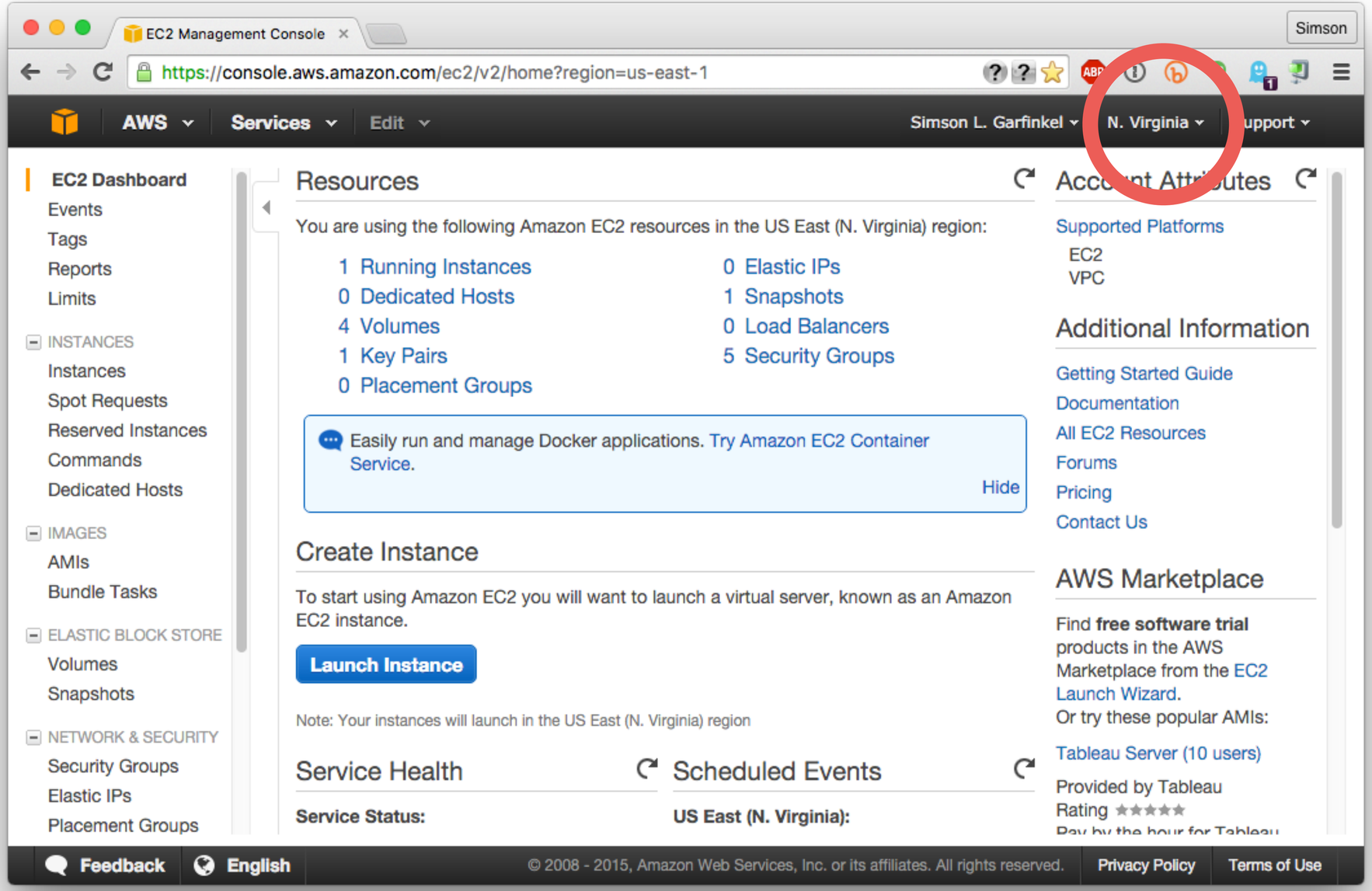
Keep watch on the "region"
You may have instances running elsewhere at Amazon…



Keep watch on the "region"
You may have instances running elsewhere at Amazon…



Keep watch on the "region"
You may have instances running elsewhere at Amazon…



If you need more RAM on EC2, you can always swap!

t2.micro instances have 1GB of physical RAM.

If you need more, but don't want to create a bigger instance, you can swap.

Here's how to create an 8GB swap file:

```
$ sudo bash
# dd if=/dev/zero of=/var/swapfile bs=1048576 count=4096
# chown 600 /var/swapfile
# mkswap /var/swapfile
# swapon /var/swapfile
# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 4  0   4248  96328  29424 236608    0    0    11   126   20   32  2  0 98  0  0
#
```

Remember — swapping slows down a system significantly.

- "You can't fake what you don't have." — Seymour Cray

EC2 Command Line Tools

```
$ aws ec2 describe-
instances
```

Amazon provides command line tools

Can be run from *any* Linux, Mac or Windows computer.

- Faster interaction than web interface.
- Can be scripted.

AWS Command Line Interface

- Run through “aws” command
- Flexible output — JSON, text, tables
- List EC2 instance: $ aws ec2 describe-instances
  - https://aws.amazon.com/cli/
  - http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

Elastic Comput Cloud CLI

- Run through 176 different ec2-* commands
- List EC2 instances: $ ec2-describe-instances
  - http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-DescribeVolumes.html

Credentials:

- Credentials kept in $HOME/.aws/ directory
- Credentials kept in AWS_USERNAME, AWS_ACCESS_KEY, AWS_SECRET_KEY environment variables.

Both are pre-installed on Amazon’s AMIs. Use the AWS CLI if possible.

Set up your environment variables and test:

AWS CLI command:

```
$ aws ec2 describe-regions
REGIONS      ec2.eu-west-1.amazonaws.com eu-west-1
REGIONS      ec2.ap-southeast-1.amazonaws.com     ap-southeast-1
REGIONS      ec2.ap-southeast-2.amazonaws.com     ap-southeast-2
REGIONS      ec2.eu-central-1.amazonaws.com       eu-central-1
REGIONS      ec2.ap-northeast-1.amazonaws.com     ap-northeast-1
REGIONS      ec2.us-east-1.amazonaws.com us-east-1
REGIONS      ec2.sa-east-1.amazonaws.com sa-east-1
REGIONS      ec2.us-west-1.amazonaws.com us-west-1
REGIONS      ec2.us-west-2.amazonaws.com us-west-2
```

(Old-style EC2- command)

```
$ ec2-describe-regions
REGION       eu-west-1 ec2.eu-west-1.amazonaws.com
REGION       ap-southeast-1    ec2.ap-southeast-1.amazonaws.com
REGION       ap-southeast-2    ec2.ap-southeast-2.amazonaws.com
REGION       eu-central-1      ec2.eu-central-1.amazonaws.com
REGION       ap-northeast-1    ec2.ap-northeast-1.amazonaws.com
REGION       us-east-1 ec2.us-east-1.amazonaws.com
REGION       sa-east-1 ec2.sa-east-1.amazonaws.com
REGION       us-west-1 ec2.us-west-1.amazonaws.com
REGION       us-west-2 ec2.us-west-2.amazonaws.com
```

EC2 has a command-line interface

Show running instances:

```
$ aws ec2 describe-instances --output text
RESERVATION             r-d9792009      376778049323
INSTANCE i-5c306beb     ami-60b6c60a                    ip-172-30-1-33.ec2.internal     running         mucha           0                       t2.micro        2015-12-02T01:47:21+0000
        us-east-1b                                              monitoring-disabled             52.90.221.164   172.30.1.33     vpc-8e73cfeb    subnet-b1de03c6 ebs
                                        hvm             xen             haLAd1447616090330              sg-15edc370     default         false           arn:aws:iam::376778049323:instance-profile/
MyWebApplication
BLOCKDEVICE             /dev/xvda       vol-8f73a76c    2015-11-15T19:34:54.000Z        true
NIC     eni-7cd9c431    subnet-b1de03c6 vpc-8e73cfeb    376778049323    in-use          172.30.1.33                     true
NICATTACHMENT           eni-attach-9d6cc676             0               attached        2015-11-15T14:34:50-0500        true
NICASSOCIATION          52.90.221.164   amazon          172.30.1.33
GROUP   sg-15edc370     default
PRIVATEIPADDRESS        172.30.1.33
TAG     instance        i-5c306beb      Name            Persistent EC2
RESERVATION             r-d05e523a      376778049323
INSTANCE i-eba98616     ami-1ecae776                    ip-172-30-1-209.ec2.internal    stopped         mucha           0                       t2.micro        2015-05-08T21:06:32+0000
        us-east-1b                                              monitoring-disabled                             172.30.1.209    vpc-8e73cfeb    subnet-b1de03c6 ebs
                                        hvm             xen             ESeOf1431119191963              sg-15edc370     default         false
BLOCKDEVICE             /dev/xvda       vol-8ff91561    2015-05-08T21:06:37.000Z        true
NIC     eni-d79fdb9f    subnet-b1de03c6 vpc-8e73cfeb    376778049323    in-use          172.30.1.209                    true
NICATTACHMENT           eni-attach-6dd55108             0               attached        2015-05-08T17:06:32-0400        true
GROUP   sg-15edc370     default
PRIVATEIPADDRESS        172.30.1.209
TAG     instance        i-eba98616      Name            TLSA Tester
RESERVATION             r-00e7e5d0      376778049323
INSTANCE i-9a48aa2c     ami-d05e75b8                    ip-172-30-1-247.ec2.internal    running         mucha           0                       t2.micro        2015-12-04T13:15:04+0000
        us-east-1b                                              monitoring-disabled             54.85.124.24    172.30.1.247    vpc-8e73cfeb    subnet-b1de03c6 ebs
                                        hvm             xen             TlrLB1448489248412              sg-e8e5ad8e     default         false           arn:aws:iam::376778049323:instance-profile/
MyWebApplication
BLOCKDEVICE             /dev/sda1       vol-e8e16e0b    2015-11-25T22:07:31.000Z        false
NIC     eni-30b8687c    subnet-b1de03c6 vpc-8e73cfeb    376778049323    in-use          172.30.1.247                    true
NICATTACHMENT           eni-attach-f5bfd91e             0               attached        2015-11-25T17:07:29-0500        true
NICASSOCIATION          54.85.124.24    amazon          172.30.1.247
GROUP   sg-e8e5ad8e     residual-study
PRIVATEIPADDRESS        172.30.1.247
PRIVATEIPADDRESS        172.30.1.35
TAG     instance        i-9a48aa2c      Name            Reminance Study
RESERVATION             r-ba2ffc90      376778049323
INSTANCE i-8e0b7f64     ami-ba13abd2                    ip-172-30-1-89.ec2.internal     stopped         windows1        0                       t2.micro        2015-04-26T14:40:59+0000
        us-east-1b                                              windows                 monitoring-disabled                             172.30.1.89     vpc-8e73cfeb    subnet-b1de03c6 ebs
                                        hvm             xen             PEYEX1416103617266              sg-15edc370     default         false
BLOCKDEVICE             /dev/sda1       vol-65202e2d    2014-11-16T02:07:01.000Z        true
NIC     eni-9f546be9    subnet-b1de03c6 vpc-8e73cfeb    376778049323    in-use          172.30.1.89                     true
NICATTACHMENT           eni-attach-30898753             0               attached        2014-11-15T21:06:57-0500        true
GROUP   sg-15edc370     default
PRIVATEIPADDRESS        172.30.1.89
TAG     instance        i-8e0b7f64      Name            Quicken
[Dance ~ 10:34:10]$
```

Use “help” to get help

```
$ aws ec2 describe-instances help


NAME
       describe-instances -

DESCRIPTION
       Describes one or more of your instances.

       If you specify one or more instance IDs, Amazon EC2 returns information
       for those instances. If you do not specify  instance  IDs,  Amazon  EC2
       returns  information  for  all  relevant  instances.  If you specify an
       instance ID that is not valid, an error is returned. If you specify  an
       instance  that  you  do  not  own,  it  is not included in the returned
       results.

       Recently terminated instances might appear  in  the  returned  results.
       This interval is usually less than one hour.

       describe-instances  is a paginated operation. Multiple API calls may be
       issued in order to retrieve the entire data set  of  results.  You  can
       disable pagination by providing the --no-paginate argument.  When using
       --output text and the --query argument on  a  paginated  response,  the
       --query  argument  must  extract data from the results of the following
       query expressions: Reservations

SYNOPSIS
            describe-instances
          [--dry-run | --no-dry-run]
          [--instance-ids <value>]
          [--filters <value>]
          [--cli-input-json <value>]
          [--starting-token <value>]
          [--page-size <value>]
          [--max-items <value>]
          [--generate-cli-skeleton]
```

$ ec2-describe-instance-status — see what's running

```
$ aws ec2 describe-instance-status --output=text
INSTANCESTATUSES    us-east-1b    i-5c306beb
INSTANCESTATE       16            running
INSTANCESTATUS      ok
DETAILS             reachability  passed
SYSTEMSTATUS        ok
DETAILS             reachability  passed
INSTANCESTATUSES    us-east-1b    i-9a48aa2c
INSTANCESTATE       16            running
INSTANCESTATUS      ok
DETAILS             reachability  passed
SYSTEMSTATUS        ok
DETAILS             reachability  passed
$
```

Change output format:

```
$ aws ec2 describe-instance-status --output=table
-------------------------------------
|       DescribeInstanceStatus       |
+------------------+--------------+--+
|         InstanceStatuses          ||
+------------------+--------------+--+
| AvailabilityZone | InstanceId   ||
+------------------+--------------+--+
|  us-east-1b      |  i-5c306beb  ||
+------------------+--------------+--+
||          InstanceState          ||
|+------------+-------------------+||
||    Code    |       Name        ||
|+------------+-------------------+||
||  16        |  running          ||
|+------------+-------------------+||
||         InstanceStatus         ||
|+------------------+-------------+||
||  Status          |  ok         ||
|+------------------+-------------+||
|||            Details            |||
||+-----------------+-------------+||
|||       Name      |  Status     |||
||+-----------------+-------------+||
|||  reachability   |  passed     |||
||+-----------------+-------------+||
```

JSON output is more useful for scripting

```
$ aws ec2 describe-instance-status --output=json
{
    "InstanceStatuses": [
        {
            "InstanceId": "i-5c306beb",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "AvailabilityZone": "us-east-1b",
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "InstanceStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            }
        },
        {
            "InstanceId": "i-9a48aa2c",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
...
```

$ aws ec2 describe-instances —instance-ids=*instance-id*

```
$ aws ec2 describe-instances --instance-ids i-5c306beb --output=table
-------------------------------------------------------------------------------
|                            DescribeInstances                                |
+-----------------------------------------------------------------------------+
||                              Reservations                                 ||
|+-----------------------------------+-----------------------------------+|
||  OwnerId                          |  376778049323                     ||
||  ReservationId                    |  r-d9792009                       ||
|+-----------------------------------+-----------------------------------+|
|||                               Instances                               |||
||+-------------------------------+---------------------------------------+||
|||  AmiLaunchIndex               |  0                                    |||
|||  Architecture                 |  x86_64                               |||
|||  ClientToken                  |  haLAd1447616090330                   |||
|||  EbsOptimized                 |  False                                |||
|||  Hypervisor                   |  xen                                  |||
|||  ImageId                      |  ami-60b6c60a                         |||
|||  InstanceId                   |  i-5c306beb                           |||
|||  InstanceType                 |  t2.micro                             |||
|||  KeyName                      |  mucha                                |||
|||  LaunchTime                   |  2015-12-02T01:47:21.000Z             |||
|||  PrivateDnsName               |  ip-172-30-1-33.ec2.internal          |||
|||  PrivateIpAddress             |  172.30.1.33                          |||
|||  PublicDnsName                |                                       |||
|||  PublicIpAddress              |  52.90.221.164                        |||
|||  RootDeviceName               |  /dev/xvda                            |||
|||  RootDeviceType               |  ebs                                  |||
|||  SourceDestCheck              |  True                                 |||
|||  StateTransitionReason        |                                       |||
|||  SubnetId                     |  subnet-b1de03c6                      |||
|||  VirtualizationType           |  hvm                                  |||
|||  VpcId                        |  vpc-8e73cfeb                         |||
||+-------------------------------+---------------------------------------+||
      ...
```

Per-instance metadata: Letting the instance know what it is

HTTP API:

```
$ curl http://169.254.169.254/latest/meta-data/instance-id
i-5c306beb$

$ aws_instance=$(wget -q -O- http://169.254.169.254/latest/meta-data/instance-id)
$ aws_region=$(wget -q -O- http://169.254.169.254/latest/meta-data/hostname)
$ echo $aws_instance $aws_region
i-5c306beb ip-172-30-1-33.ec2.internal
$
```

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

ec2-metadata:

```
$ ec2-metadata  -i
instance-id: i-5c306beb
$ ec2-metadata  -i| awk '{print $2;}'
i-5c306beb
```

EC2 Pricing

$$

Current EC2 pricing…

Region: US East (N. Virginia)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.micro	1	Variable	1	EBS Only	$0.013 per Hour
t2.small	1	Variable	2	EBS Only	$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	$0.052 per Hour
t2.large	2	Variable	8	EBS Only	$0.104 per Hour
m4.large	2	6.5	8	EBS Only	$0.126 per Hour
m4.xlarge	4	13	16	EBS Only	$0.252 per Hour
m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	$1.008 per Hour
m4.10xlarge	40	124.5	160	EBS Only	$2.52 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	$0.067 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	$0.133 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	$0.266 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	$0.532 per Hour

Costing your instance…

Amazon bills AMI's at cost per hour…

… but you need to think about performance per dollar.

Performance is determined by:

- Location
- Total Memory
- Memory per core
- Storage
- Network
- Operating System

Will you get more work done with one m4.2xlarge or 2 m4.4xlarge?

m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	$1.008 per Hour

You can save a *lot* of money with "spot instances"

Fixed vs. Spot pricing, Nov 29, 2015:

m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances (i) 1 Launch into Auto Scaling Group (i)

Purchasing option (i) ☑ Request Spot instances

Current price (i)

us-east-1a	0.3195
us-east-1b	0.1427
us-east-1c	0.1727
us-east-1e	0.0992

Maximum price (i) $ (e.g. 0.045 = 4.5 cents/hour)

Case Study: Cycle Computing's 4096-core EC2 Cluster (2011)

Key questions:

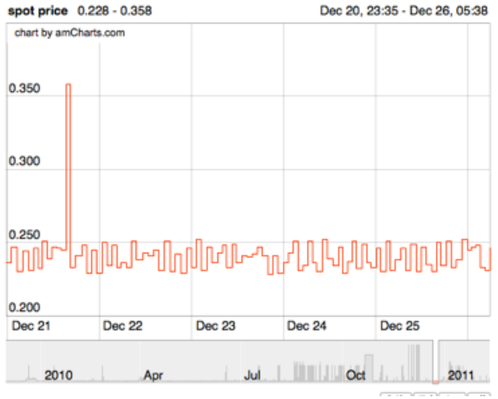
- Could they get 4096 cores from EC2 reliably? (512 c1.xlarge instances, 8 virtual cores each)
- Could the configuration management software keep it up?
- Would their scheduler scale?

What they did:

- Asked Amazon to raise instance limit.
- Ordered new instances in batches of 64.

What they found:

- Spot prices didn't go up even when instances were not available!



More info:

- http://cyclecomputing.com/blog/cyclecloud-4096-core-cluster/

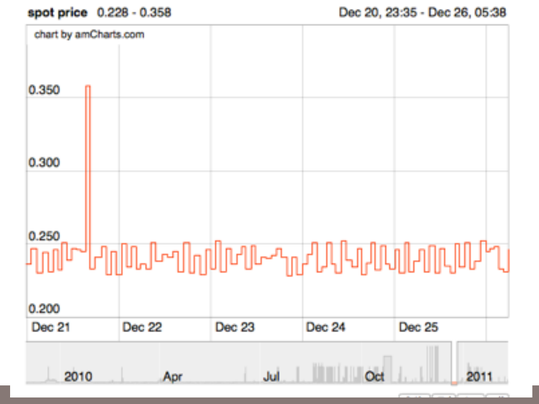
To Procure 512 large instances, build in batches of 64.

Cycle Computing requested instances in batches of 64

Time (EST)	Instances Requested	AZ Requested	Actual AZ	Notes
13:08:00	64	us-east-1a	us-east-1d	1 DOA; 1 disk failure
13:15:00	65	us-east-1b	us-east-1b	1 DOA
13:25:00	64	us-east-1c	us-east-1d	3 not reachable
13:30:00	64	us-east-1b	us-east-1b	
13:36:00	64	us-east-1a	us-east-1b	1 DOA
13:42:00	64	us-east-1d	us-east-1b	
14:05:00	64	us-east-1c	us-east-1b	
14:13:00	81	us-east-1b	us-east-1b	1 DOA
14:18:00	1	us-east-1b	us-east-1b	

- Failure rate: 0.75 - 1.00% (remember, this is 2011 data)

Spot pricing was not impacted, even when instances were not available



(I'm not sure I believe what they claim, as they don't show their allocations on the graph.)

Price to build a 4096-core HPC system:

Article's quoted prices:

Item	Price (AWS + CycleCloud)	Quantity	Cost/hr
C1.XLarge instances	$0.816/instance-hr	512 Instances x 8-cores	$417.79
Filer	$0.138 / TB-hour (approx.)	1 TB	$0.14
		TOTAL per Hour	$417.93

Today's prices, sans CycleCloud:

Item	Cores	Instances Required	GB	GB/core	Cost/Hour	Total Cost/hour
C1.XLarge	8	512	7	0.875	$0.52	$266.24
m4.10xlarge	40	102	160	4	$2.52	$257.04

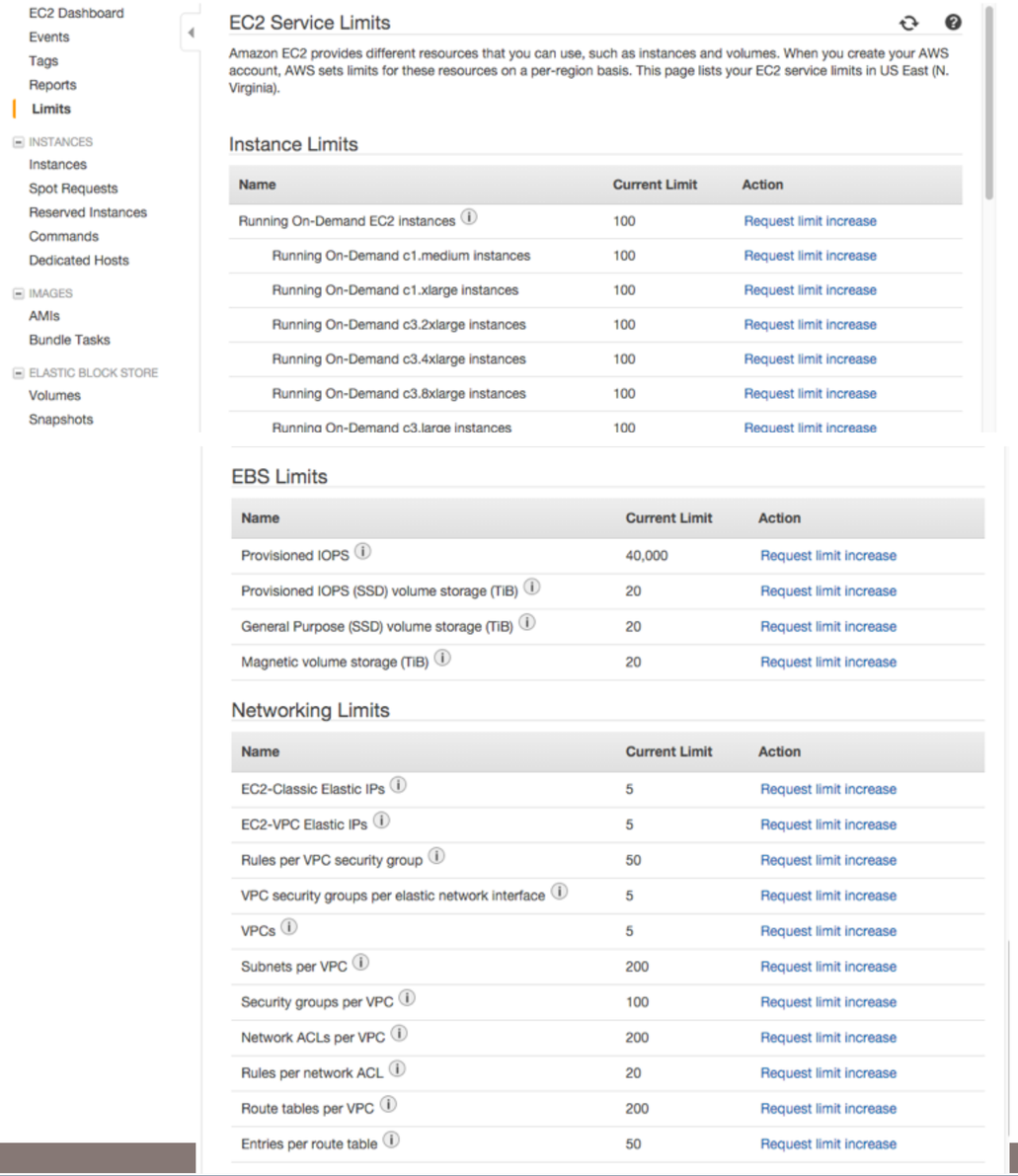
EC2 Service "Limits"

Limits are by:

- Instance type
- Storage type
- Networking type

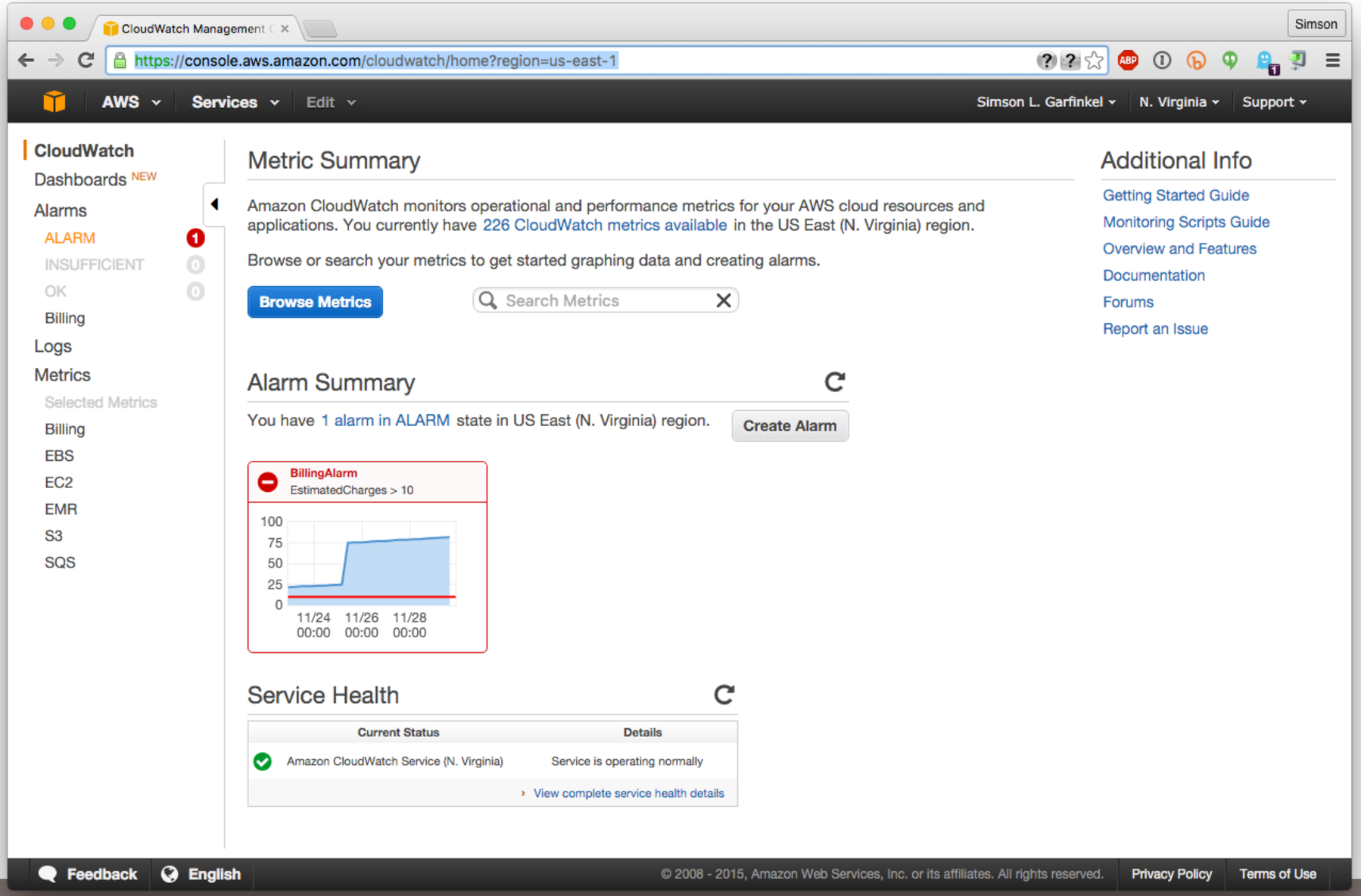
You can ask AWS to increase your limits.

- Don't do it for this course!

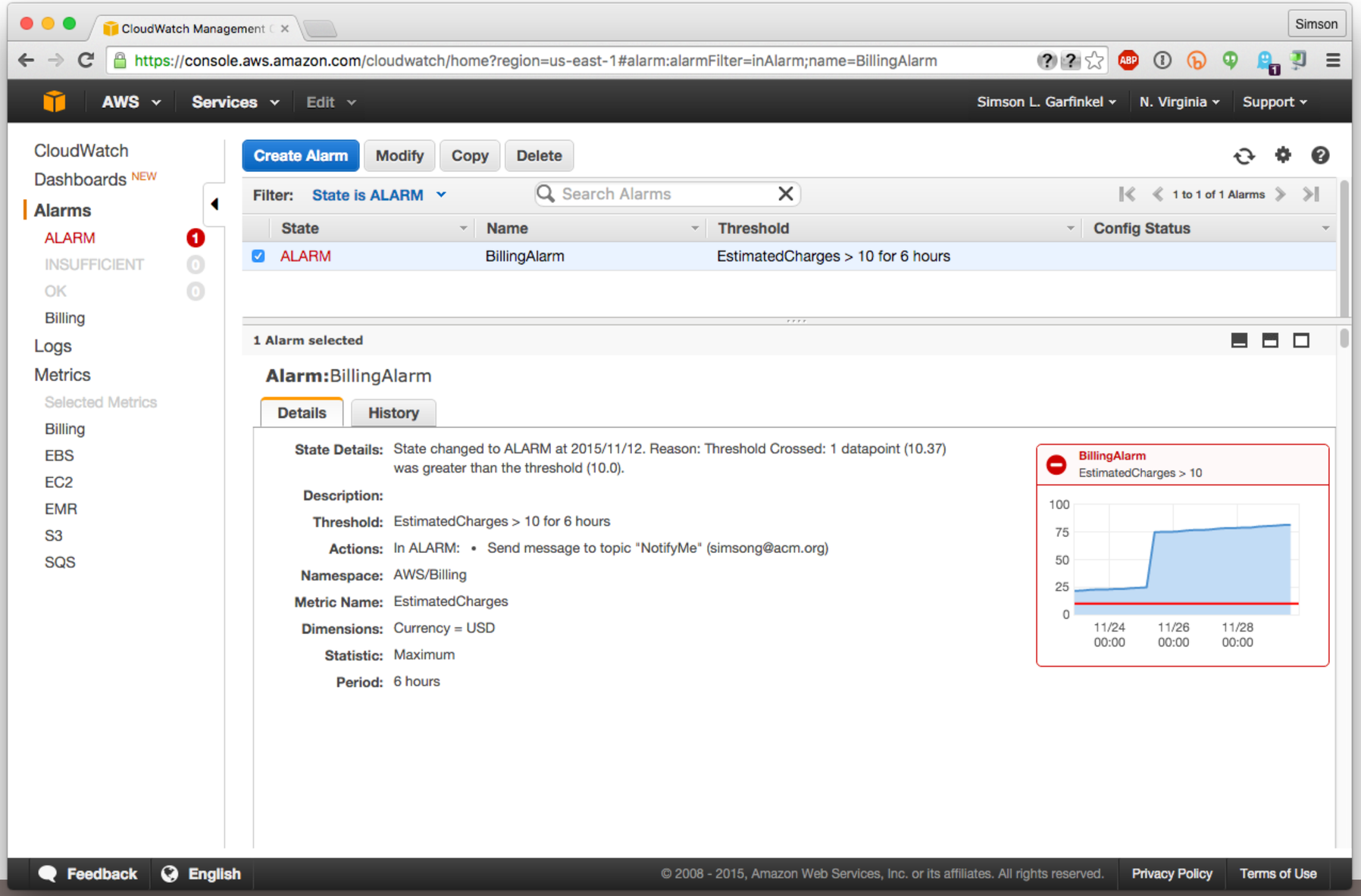


Amazon CloudWatch

CloudWatch alerts you if something is getting out of control.



You should set up alarms!



AWS Notifications

To: Simson L. Garfinkel

ALARM: "BillingAlarm" in US - N. Virginia

November 12, 2015 at 7:42 AM

Archive - Google (All Mail)

AN

You are receiving this email because your estimated charges are greater than the limit you set for the alarm "BillingAlarm" in AWS Account 376778049323.

The alarm limit you set was $ 10.00 USD. Your total estimated charges accrued for this billing period are currently $ 10.37 USD as of Thursday 12 November, 2015 12:42:12 UTC. The actual charges you will be billed in this statement period may differ from the charges shown on this notification. For more information, view your estimated bill at:
https://console.aws.amazon.com/billing/home#/bill?year=2015&month=11

More details about this alarm are provided below:

Amazon CloudWatch Alarm "BillingAlarm" in the US - N. Virginia region has entered the ALARM state, because "Threshold Crossed: 1 datapoint (10.37) was greater than the threshold (10.0)." at "Thursday 12 November, 2015 12:42:12 UTC".

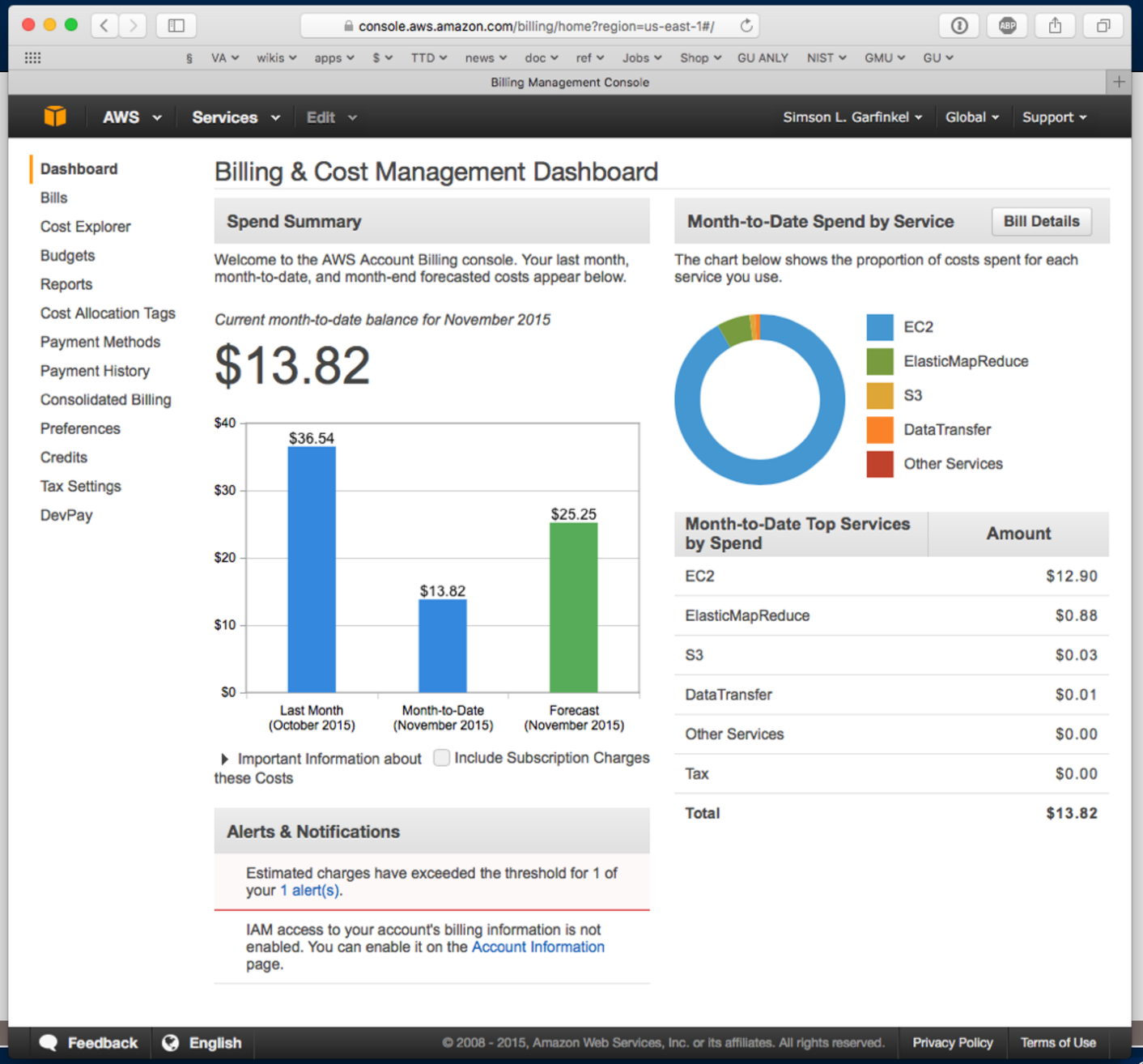
View this alarm in the AWS Management Console:
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#s=Alarms&alarm=BillingAlarm

Alarm Details:
- Name: BillingAlarm
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint (10.37) was greater than the threshold (10.0).
- Timestamp: Thursday 12 November, 2015 12:42:12 UTC
- AWS Account: 376778049323

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 10.00 for 21600 seconds.

Monitored Metric:
- MetricNamespace: AWS/Billing
- MetricName: EstimatedCharges
- Dimensions: [Currency = USD]
- Period: 21600 seconds
- Statistic: Maximum
- Unit: not specified

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:376778049323:NotifyMe]
- INSUFFICIENT_DATA:

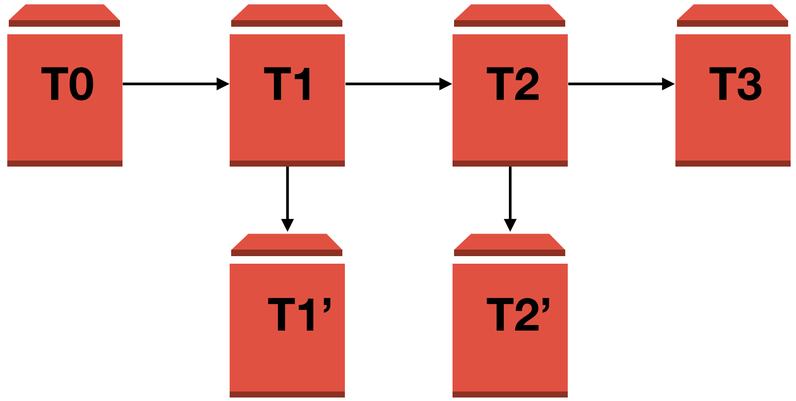

console.aws.amazon.com/billing/home?region=us-east-1#/
Billing Management Console
AWS
Services
Edit
Simson L. Garfinkel
Global
Support
Dashboard
Bills
Cost Explorer
Budgets
Reports
Cost Allocation Tags
Payment Methods
Payment History
Consolidated Billing
Preferences
Credits
Tax Settings
DevPay
Billing & Cost Management Dashboard
Spend Summary
Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.
Current month-to-date balance for November 2015
$13.82
$40
$30
$20
$10
$0
$36.54
$13.82
$25.25
Last Month (October 2015)
Month-to-Date (November 2015)
Forecast (November 2015)
Important Information about these Costs
Include Subscription Charges
Alerts & Notifications
Estimated charges have exceeded the threshold for 1 of your 1 alert(s).
IAM access to your account's billing information is not enabled. You can enable it on the Account Information page.
Month-to-Date Spend by Service
Bill Details
The chart below shows the proportion of costs spent for each service you use.
EC2
ElasticMapReduce
S3
DataTransfer
Other Services
Month-to-Date Top Services by Spend
Amount
EC2 $12.90
ElasticMapReduce $0.88
S3 $0.03
DataTransfer $0.01
Other Services $0.00
Tax $0.00
Total $13.82
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Amazon EBS

EBS: Virtual Disk Volumes

EBS volumes:

- Created automatically when EC2 instance starts up.
- Snapshots on the fly.



Options:

- Magnetic or SSD
- Destroy or persist on instance termination
- Not Encrypted / Encrypted
- Provisioned IOPS

Uses:

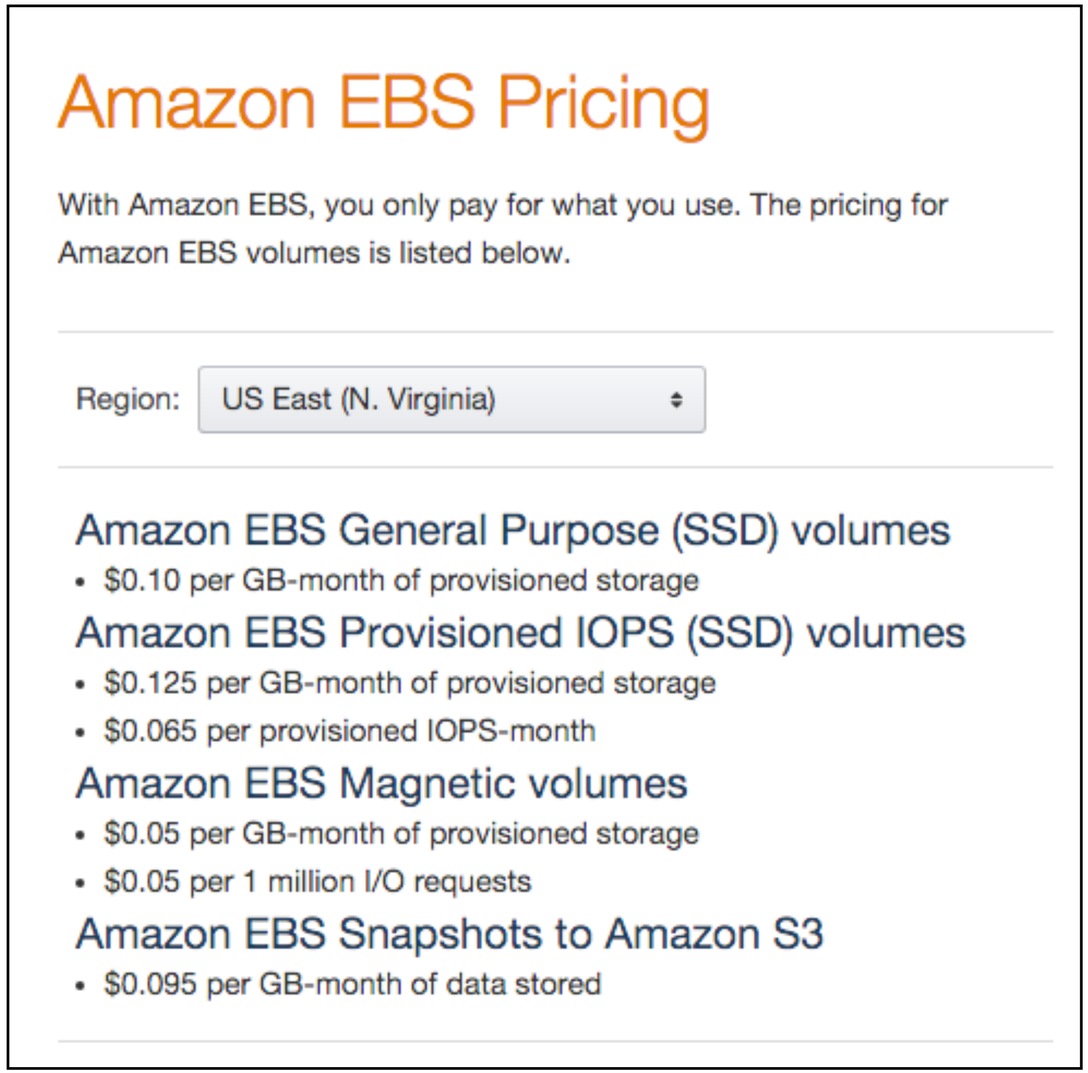
- Boot drives
- Read-only drives to share static databases. (Make 1 TB drive and mount)
- Database drives for MySQL, etc.. (But you should use Amazon's managed service.)

EBS offers three classes of service.

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
Use cases	• System boot volumes • Virtual desktops • Small to medium sized databases • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume • Large database workloads, such as: ○ MongoDB ○ Microsoft SQL Server ○ MySQL ○ PostgreSQL ○ Oracle	• Cold workloads where data is infrequently accessed • Scenarios where the lowest storage cost is important
Volume size	1 GiB – 16 TiB	4 GiB – 16 TiB	1 GiB – 1 TiB
Maximum throughput	160 MiB/s	320 MiB/s	40–90 MiB/s
IOPS performance	Baseline performance of 3 IOPS/GiB (up to 10,000 IOPS) with the ability to burst to 3,000 IOPS for volumes under 1,000 GiB.	Consistently performs at provisioned level, up to 20,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS
API and CLI volume name	`gp2`	`io1`	`standard`

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

Pricing — you are probably best off with SSD General Purpose.



Amazon EBS Pricing

With Amazon EBS, you only pay for what you use. The pricing for Amazon EBS volumes is listed below.

Region: US East (N. Virginia)

Amazon EBS General Purpose (SSD) volumes

- $0.10 per GB-month of provisioned storage

Amazon EBS Provisioned IOPS (SSD) volumes

- $0.125 per GB-month of provisioned storage
- $0.065 per provisioned IOPS-month

Amazon EBS Magnetic volumes

- $0.05 per GB-month of provisioned storage
- $0.05 per 1 million I/O requests

Amazon EBS Snapshots to Amazon S3

- $0.095 per GB-month of data stored

EBS volumes can be created and used for: extra storage, sharing data

Each EBS volume has:

- Size — e.g. 40GB
- Name — e.g. vol-65202e2d
- Region / AvailabilityZone — e.g. us-east-1 / us-east-1b
- Attributes — e.g. CreateTime, Encrypted, Iops,

Volumes can be mounted:

- read/write on a single instance
- read-only on multiple instances

Create and share an instance:

```
$ aws ec2 create-volume --size 10 --availability-zone us-east-1a
You must specify a region. You can also configure your region by running "aws configure".
$ aws ec2 create-volume --size 10 --region us-east-1 --availability-zone us-east-1b
{
    "AvailabilityZone": "us-east-1b",
    "Encrypted": false,
    "VolumeType": "standard",
    "VolumeId": "vol-95cab176",
    "State": "creating",
    "SnapshotId": "",
    "CreateTime": "2015-12-05T18:55:28.052Z",
    "Size": 10
}
$
```

Attach the EBS volume to your VM (Be sure EBS is in same region & availability zone)

First get a volume...

```
$ aws_zone=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)
$ aws_instance=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
$ aws_region=$(curl -s http://169.254.169.254/latest/dynamic/instance-identity/document|grep region|awk -F\" '{print $4}')
$ aws ec2 create-volume --size 10 --region $aws_region --availability-zone $aws_zone
{
    "AvailabilityZone": "us-east-1b",
    "Encrypted": false,
    "VolumeType": "standard",
    "VolumeId": "vol-46cdb6a5",
    "State": "creating",
    "SnapshotId": "",
    "CreateTime": "2015-12-05T19:01:38.548Z",
    "Size": 10
}
$ aws ec2 attach-volume --volume-id=vol-46cdb6a5 --instance-id=$aws_instance \
  --device=/dev/sdb --region=$aws_region
{
    "AttachTime": "2015-12-05T19:02:11.541Z",
    "InstanceId": "i-5c306beb",
    "VolumeId": "vol-46cdb6a5",
    "State": "attaching",
    "Device": "/dev/sdb"
}
$
```

Now we need to make a file system...

Create a file system on the volume

```
$ sudo mkfs -t ext4 /dev/sdb
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: 681c57f0-1461-4dae-b956-032656ba82a9
Superblock backups stored on blocks:
     32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

$ sudo mount /dev/sdb /mnt/extra/
[ip-172-30-1-33 ~ 19:04:44]$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/xvda1      41151788 6506728  34544812  16% /
devtmpfs          500712      60    500652   1% /dev
tmpfs             509724       0    509724   0% /dev/shm
/dev/xvdb       10190136   23028   9626436   1% /mnt/extra

$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  40G  0 disk
└─xvda1 202:1    0  40G  0 part /
xvdb    202:16   0  10G  0 disk /mnt/extra
$
```

EBS Snapshots: Sharing between multiple systems (and users)

EBS volumes: only mounted read/write on one instance at a time.

- Most file systems don't support multiple writers from different systems.
- Weird consistency issues in a networked environment.

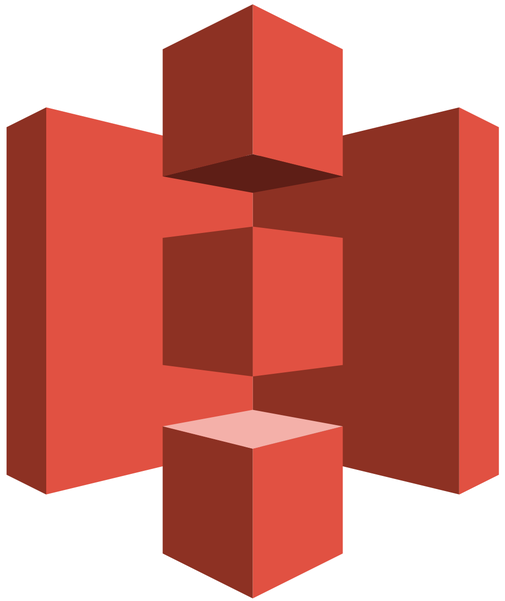
Snapshots allow:

- A single read-only volume to be mounted by many users.
- Publishing an EBS volume to others.
- Restore to a different volume.

Information on snapshots:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html
- http://angus.readthedocs.org/en/2014/amazon/using-ebs-snapshot.html

Amazon S3



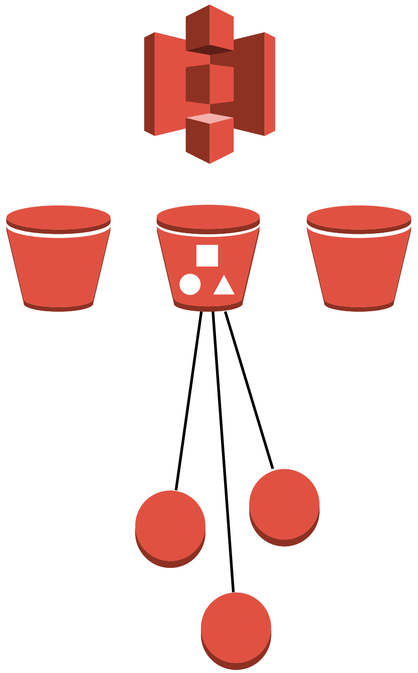
S3 is an object-based storage system

Every S3 bucket has:

- Name
- Owner
- Access permissions

Every S3 object has:

- Size
- URL
- Access permissions
  —*e.g. world readable*



Amazon S3

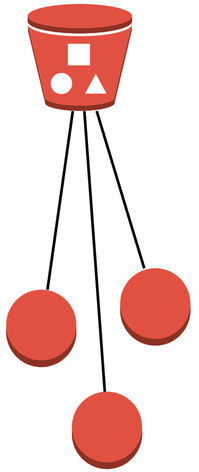
Per-user “buckets”

Objects in the bucket

Accessing S3 data

Uses of S3:

- Storing logs
- Distributing data



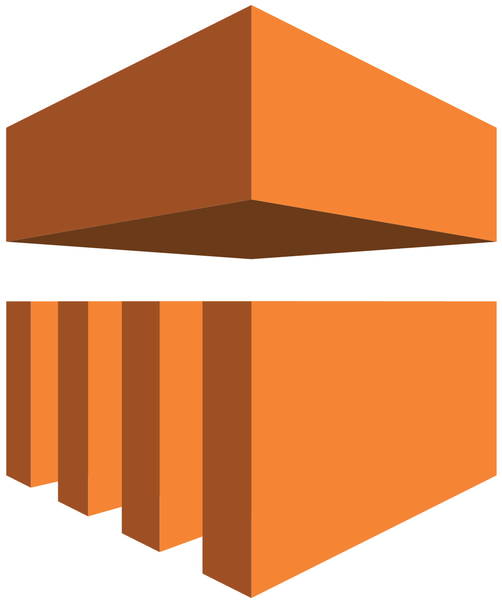
S3 with Hadoop:

- Access with Python using “boto”
- Access native objects from Hadoop using s3n://
  - —*Now replace with s3a://*
- Put HDFS inside an S3 with s3://
- Advantages:
  - —*permanence; S3 outlasts your EC2/EMR cluster*
  - —*Pay only for what you need, rather than HDFS capacity.*
- Disadvantage: no data locality. S3 data *always* moves over the network.
  - —*May not be an issue with 10g instances.*

Remember: S3 is not a file system, it’s an object storage system.

- https://wiki.apache.org/hadoop/AmazonS3

Amazon EMR



Elastic Map Reduce: Amazon's managed Hadoop cluster.

Hourly rate for every instance hour you use.

- 10-node cluster for 10 hours = 100-node cluster for 1 hour
- Higher prices for more powerful instances.
- Instance prices $0.11/hour to $0.27/hour
- EMR price is *in addition* to EC2 price.

US East is generally cheapest.

- Even cheaper with spot instances.
- Hadoop & Spark don't run on "Previous Generation" small & medium instances.

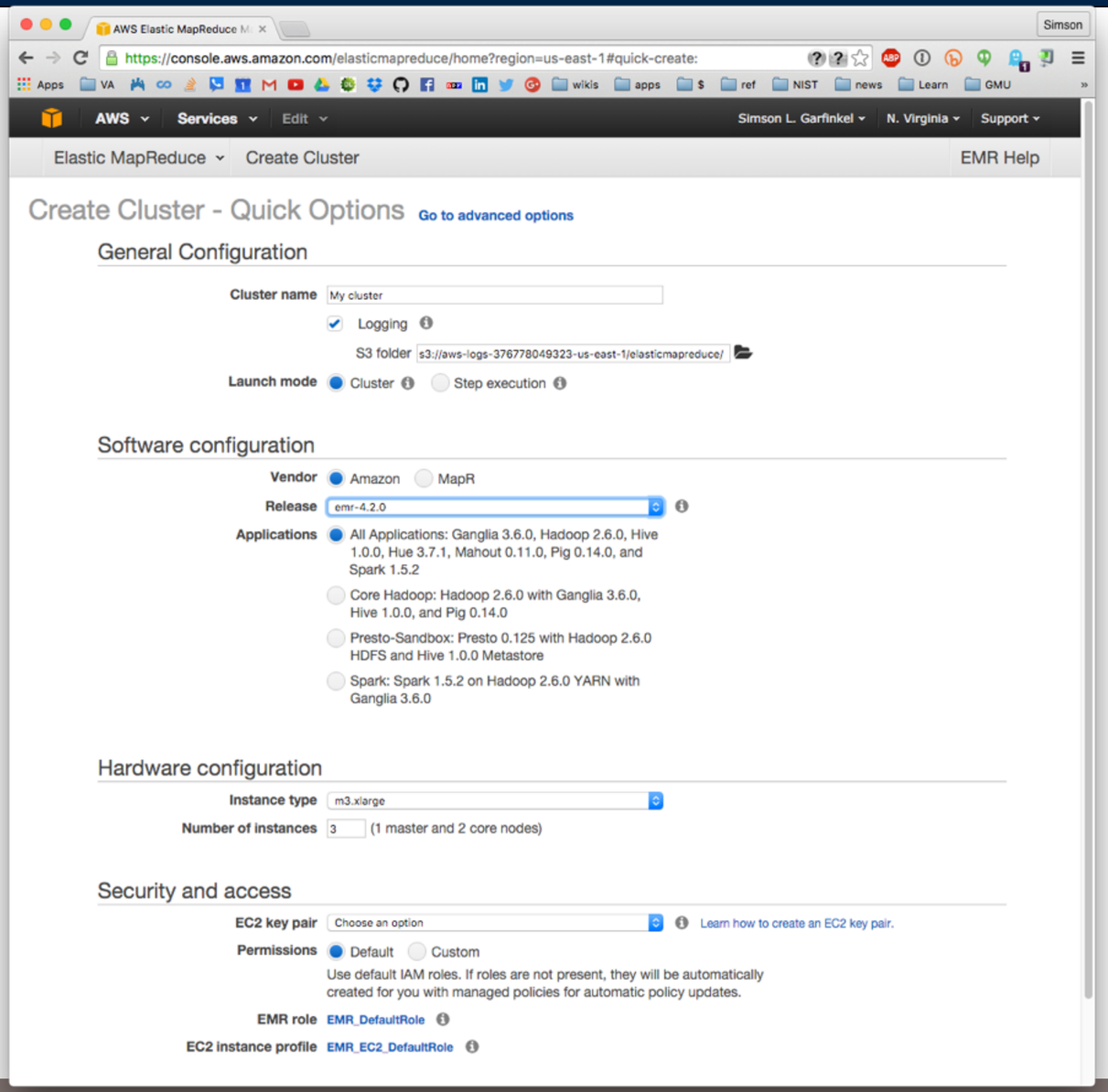
	Amazon EC2 Price	Amazon Elastic MapReduce Price
General Purpose - Current Generation		
m3.xlarge	$0.266 per Hour	$0.070 per Hour
m3.2xlarge	$0.532 per Hour	$0.140 per Hour
General Purpose - Previous Generation		
m1.small	$0.044 per Hour	$0.011 per Hour
m1.medium	$0.087 per Hour	$0.022 per Hour
m1.large	$0.175 per Hour	$0.044 per Hour
m1.xlarge	$0.350 per Hour	$0.088 per Hour
Compute Optimized - Current Generation		
c3.xlarge	$0.210 per Hour	$0.053 per Hour
c3.2xlarge	$0.420 per Hour	$0.105 per Hour
c3.4xlarge	$0.840 per Hour	$0.210 per Hour
c3.8xlarge	$1.680 per Hour	$0.270 per Hour

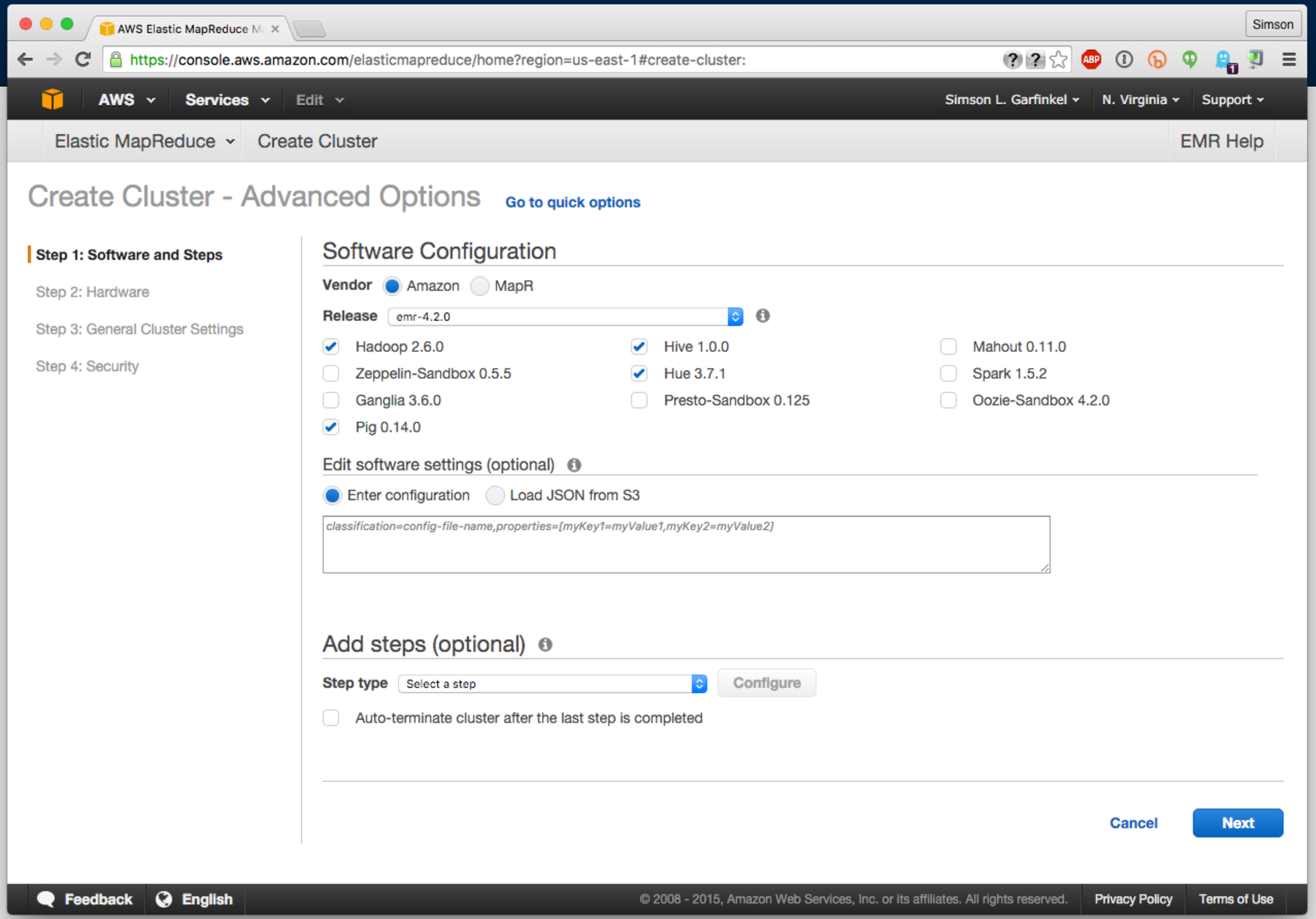
Options when creating EMR clusters

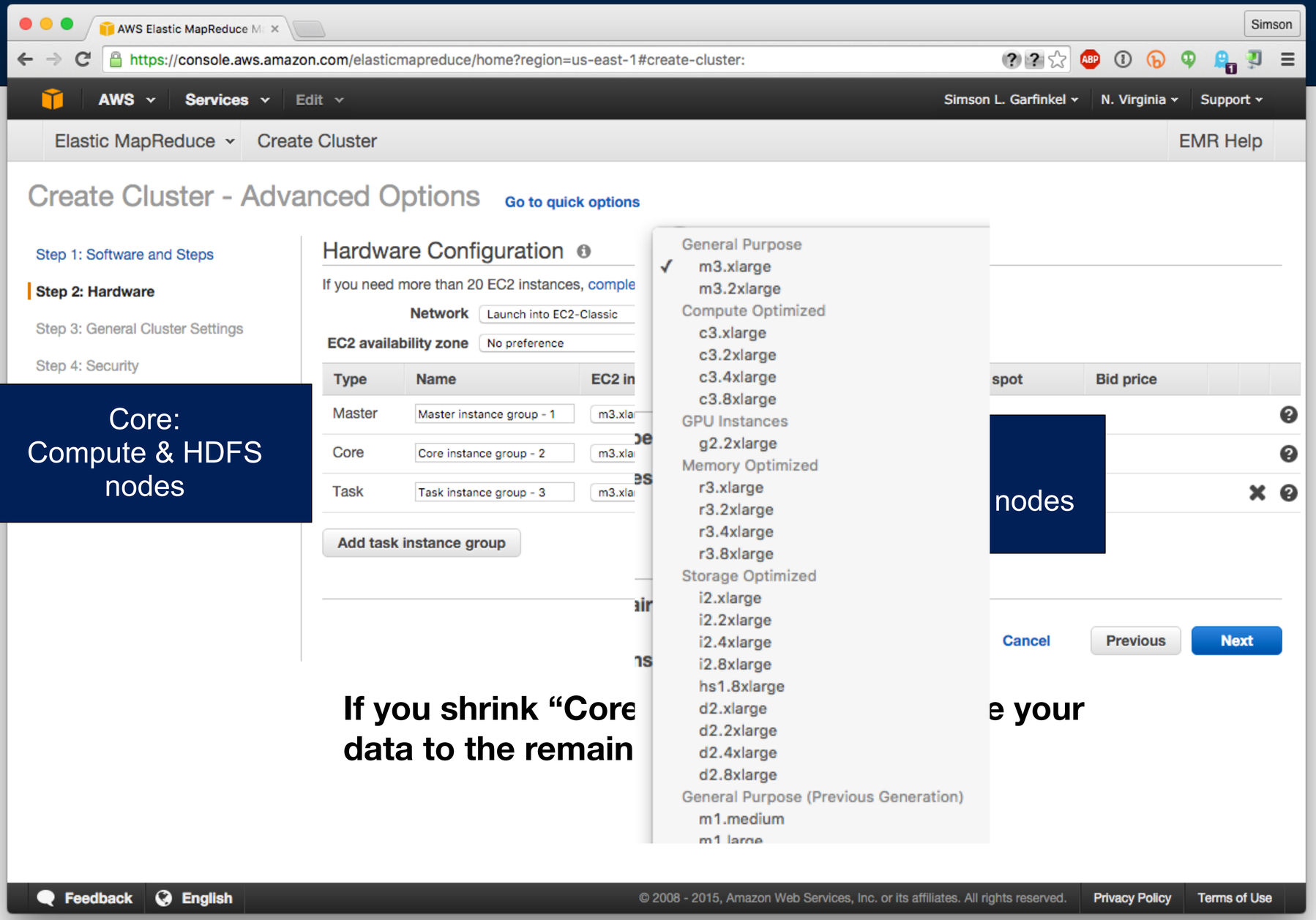
Note:

- Logging — S3 bucket
- Vendors:
  - —*Amazon & MapR*
- Instances:
  - —*Instance Type*
  - —*# of Instances*
- Security
  - —*EC2 key pair.*

But for this course, please use the "Advanced Options"



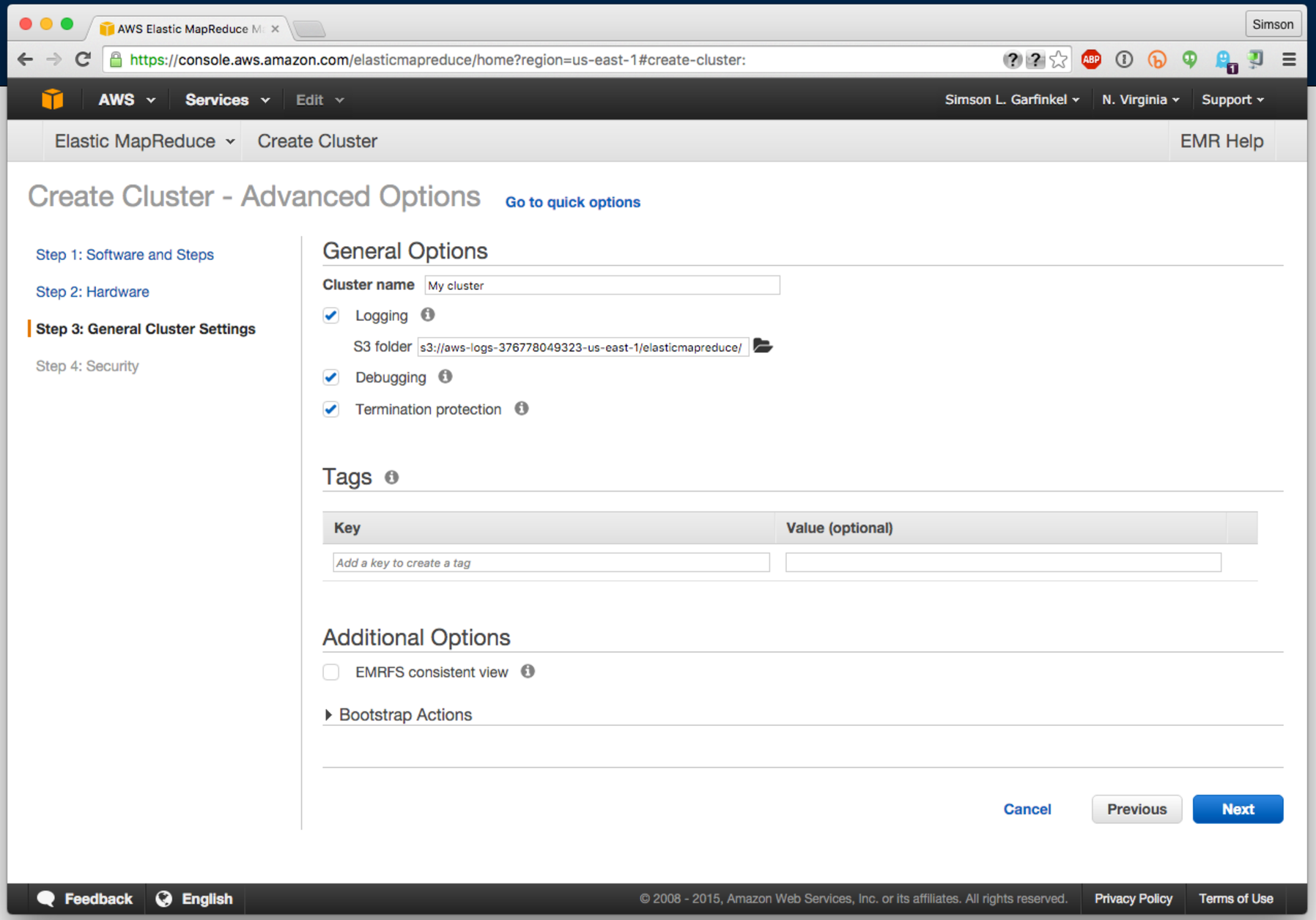

AWS Elastic MapReduce
Simson
https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#create-cluster:
AWS
Services
Edit
Simson L. Garfinkel
N. Virginia
Support
Elastic MapReduce
Create Cluster
EMR Help
Create Cluster - Advanced Options
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
Software Configuration
Vendor
Amazon
MapR
Release
emr-4.2.0
Hadoop 2.6.0
Hive 1.0.0
Mahout 0.11.0
Zeppelin-Sandbox 0.5.5
Hue 3.7.1
Spark 1.5.2
Ganglia 3.6.0
Presto-Sandbox 0.125
Oozie-Sandbox 4.2.0
Pig 0.14.0
Edit software settings (optional)
Enter configuration
Load JSON from S3
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
Add steps (optional)
Step type
Select a step
Configure
Auto-terminate cluster after the last step is completed
Cancel
Next
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



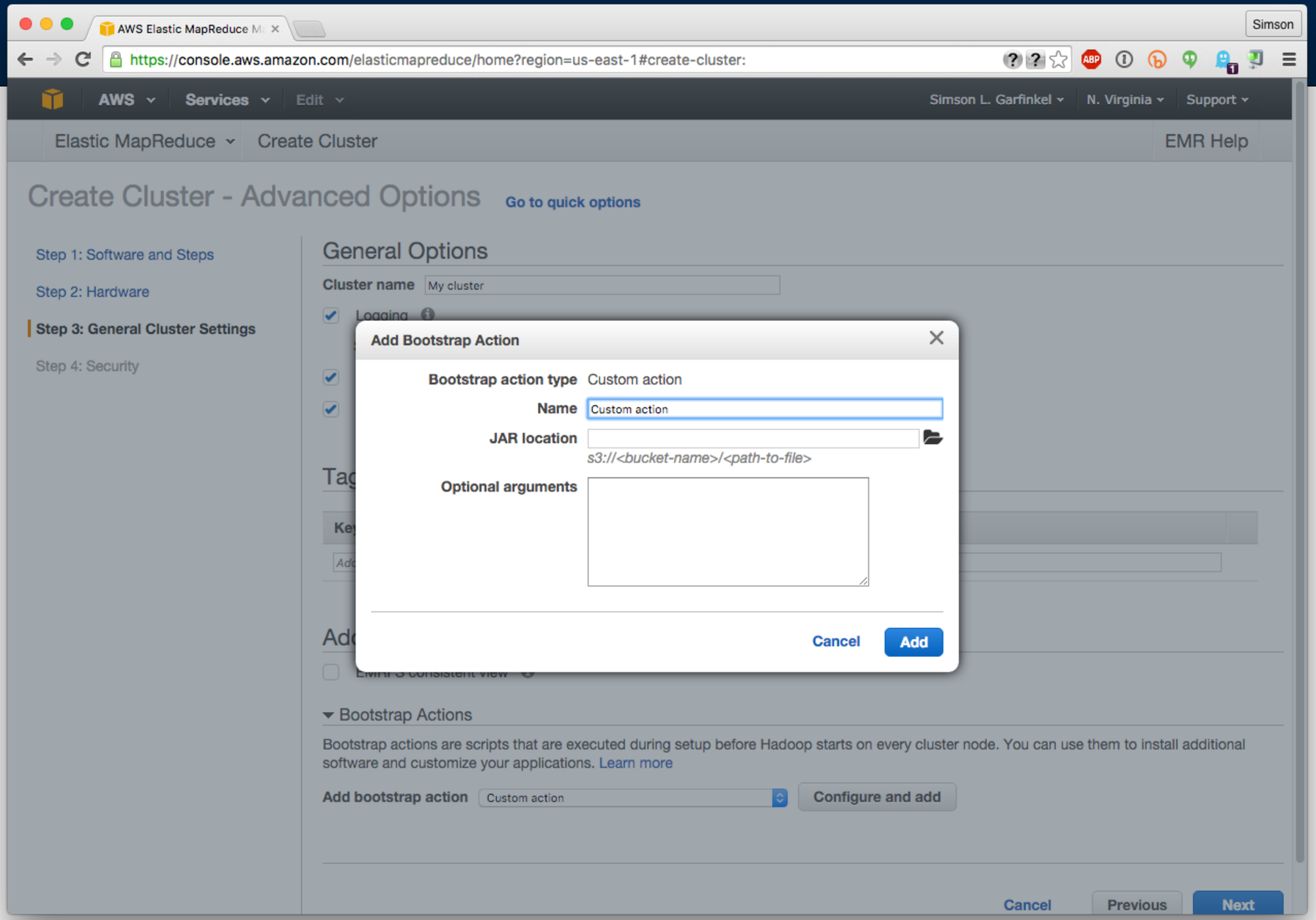
Core:
Compute & HDFS
nodes

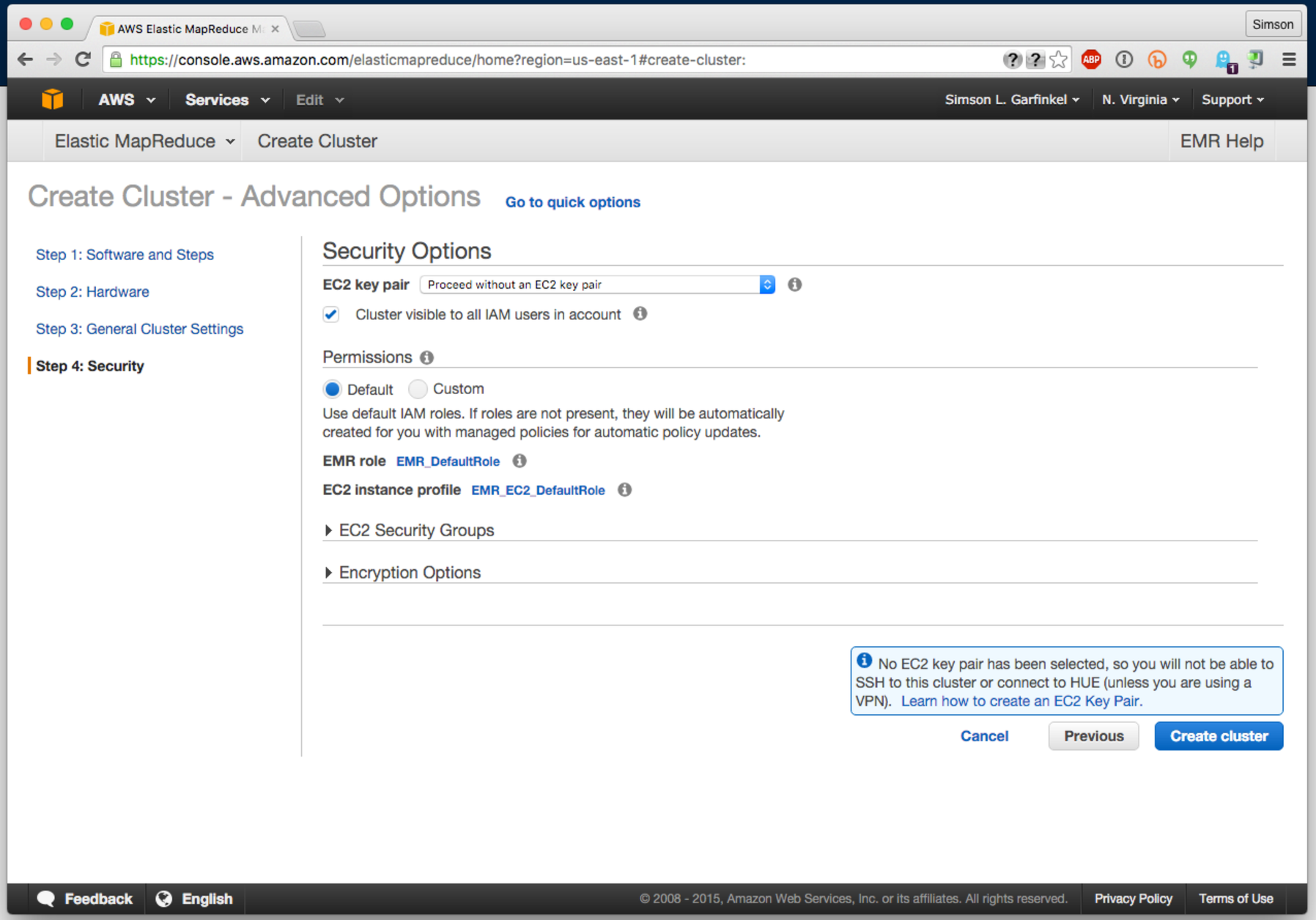
nodes

If you shrink "Core … e your data to the remain

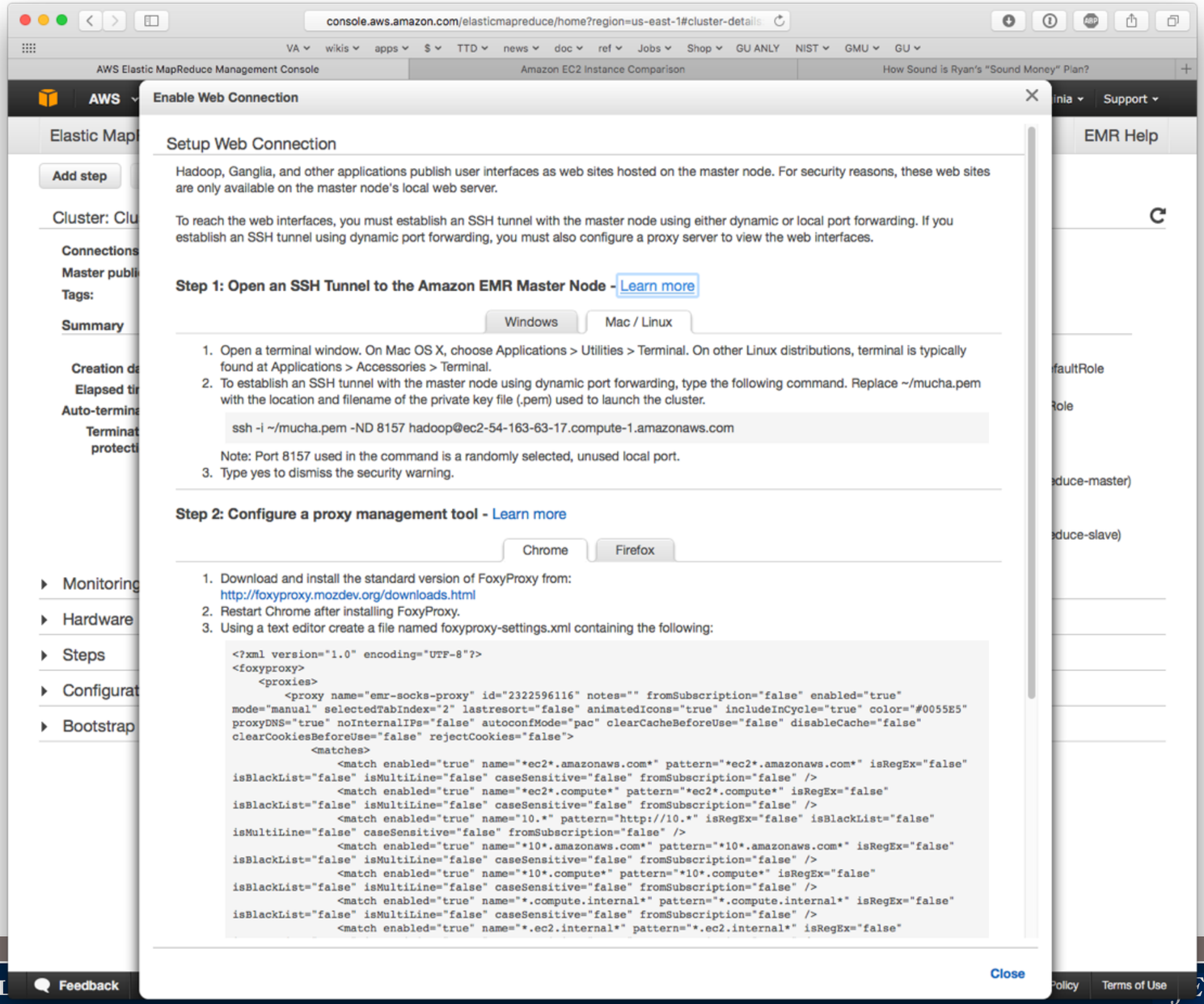

AWS Elastic MapReduce
Simson
https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#create-cluster:
AWS
Services
Edit
Simson L. Garfinkel
N. Virginia
Support
Elastic MapReduce
Create Cluster
EMR Help
Create Cluster - Advanced Options
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
General Options
Cluster name
My cluster
Logging
S3 folder
s3://aws-logs-376778049323-us-east-1/elasticmapreduce/
Debugging
Termination protection
Tags
Key
Value (optional)
Add a key to create a tag
Additional Options
EMRFS consistent view
Bootstrap Actions
Cancel
Previous
Next
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

You can create a bootstrap to pre-install mrjob.




AWS Elastic MapReduce
Simson
https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#create-cluster:
AWS
Services
Edit
Simson L. Garfinkel
N. Virginia
Support
Elastic MapReduce
Create Cluster
EMR Help
Create Cluster - Advanced Options
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
Security Options
EC2 key pair
Proceed without an EC2 key pair
Cluster visible to all IAM users in account
Permissions
Default
Custom
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.
EMR role EMR_DefaultRole
EC2 instance profile EMR_EC2_DefaultRole
EC2 Security Groups
Encryption Options
No EC2 key pair has been selected, so you will not be able to SSH to this cluster or connect to HUE (unless you are using a VPN). Learn how to create an EC2 Key Pair.
Cancel
Previous
Create cluster
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Connect with SSH tunneling and FoxProxy

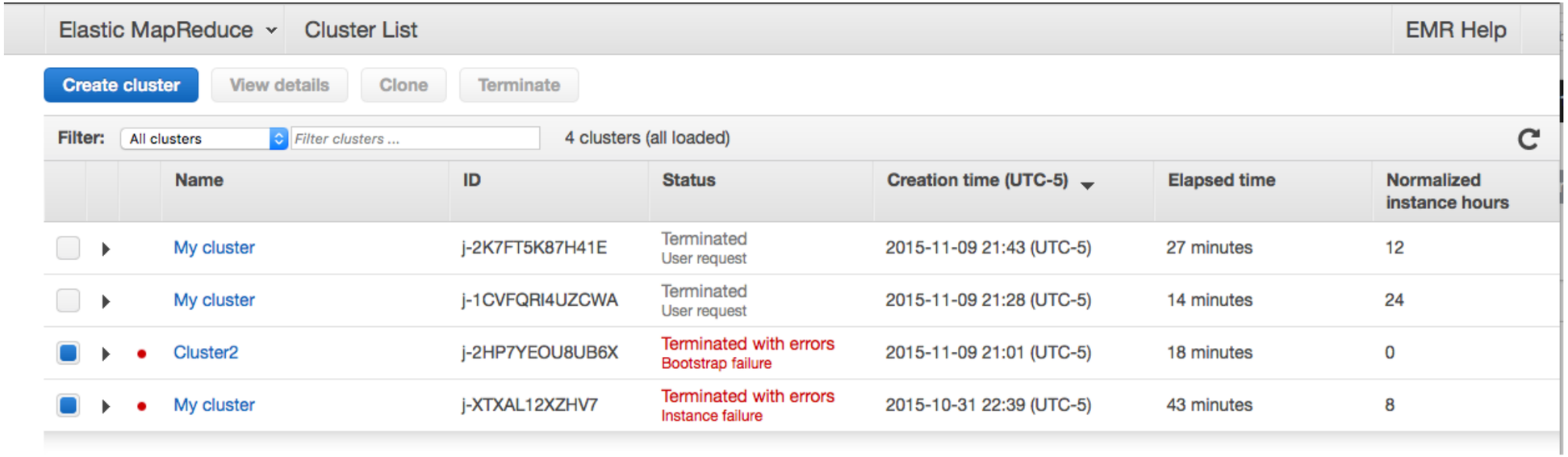


Clusters remain after termination so you can "clone" them.

Be sure to terminate in the EMR, not EC2 panel.

Remember: you lose your disk and HDFS when you terminate!

- Store things in S3 and git!



Amazon Security

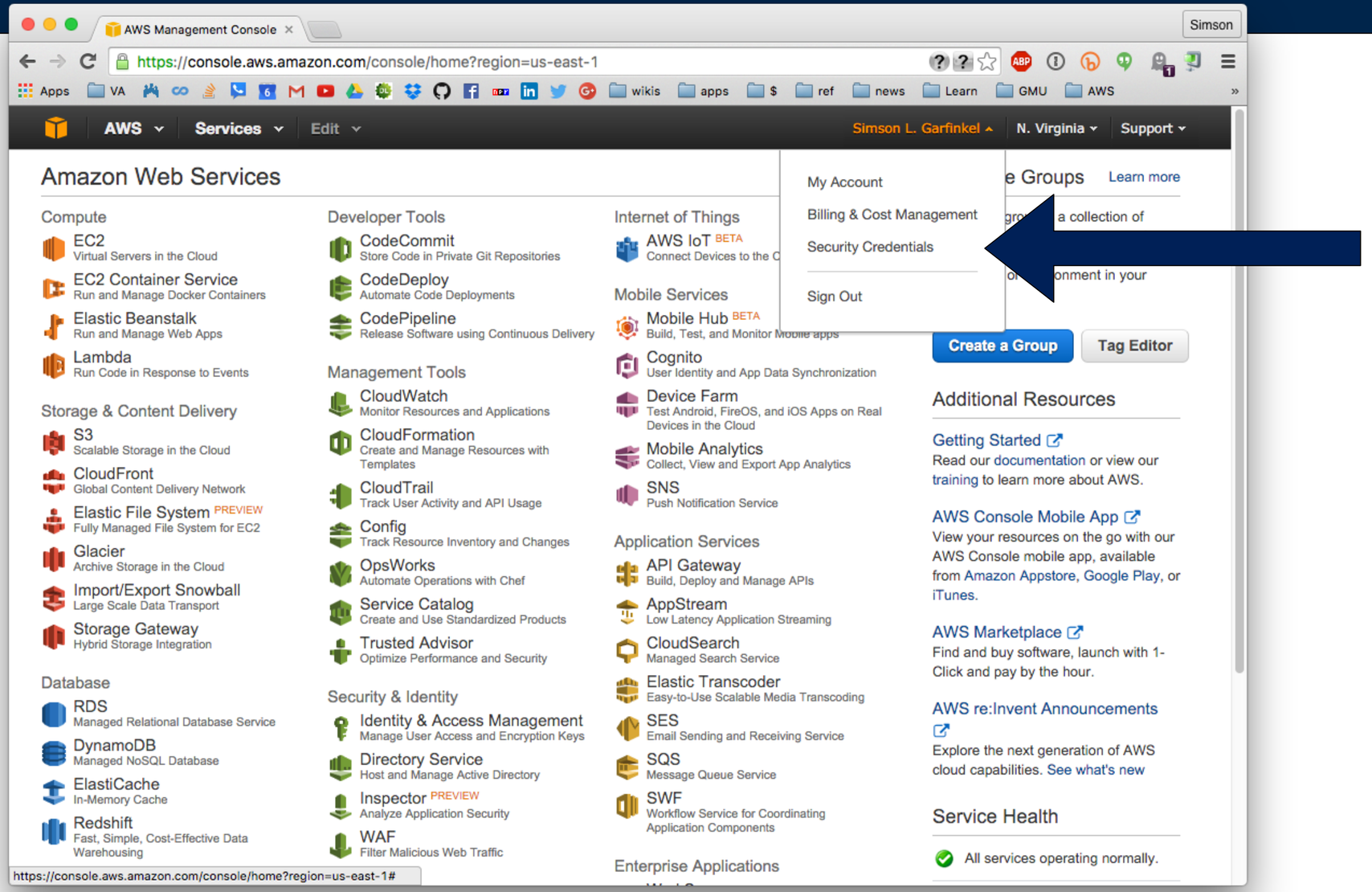
Protecting your account

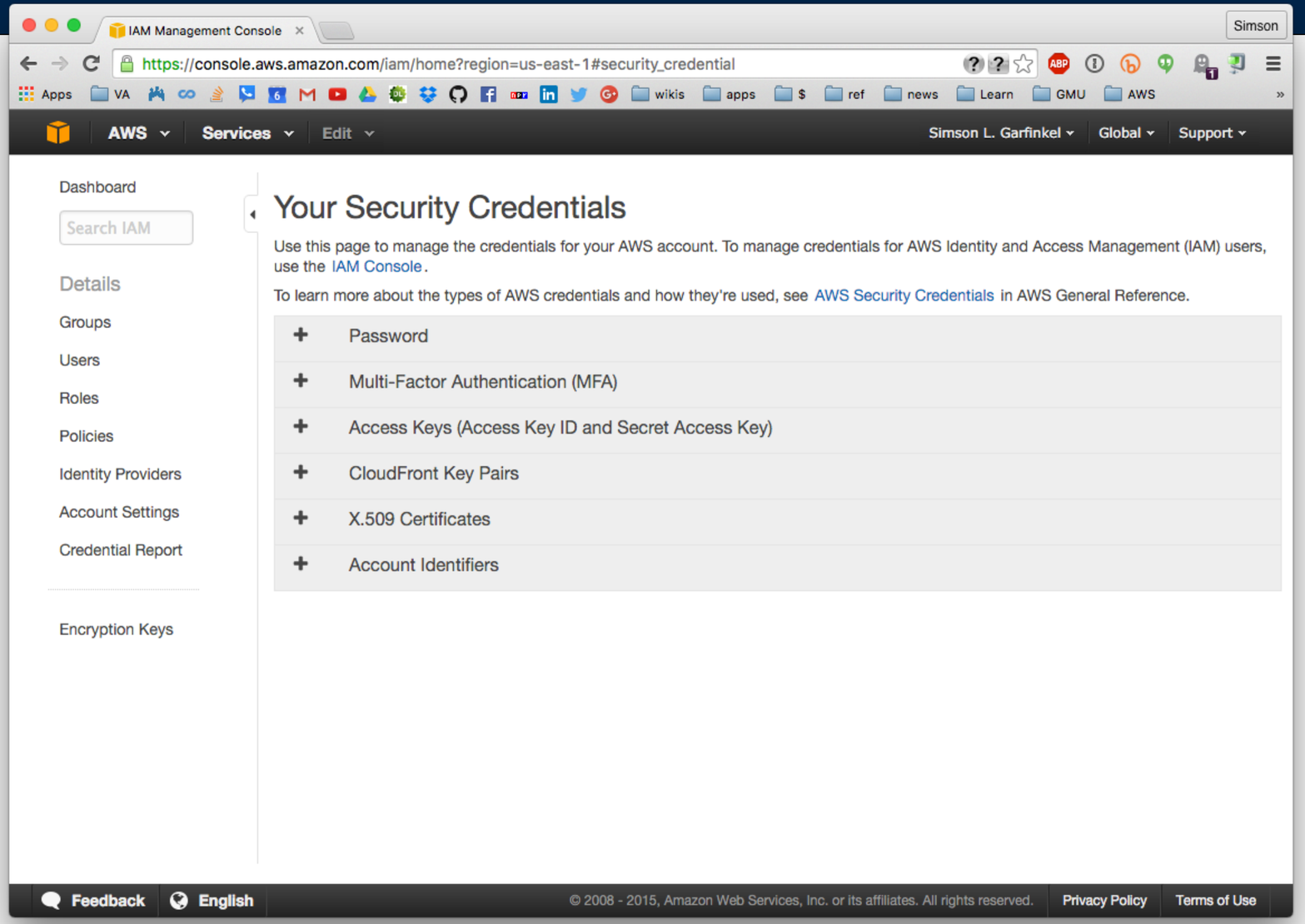
Complex password.

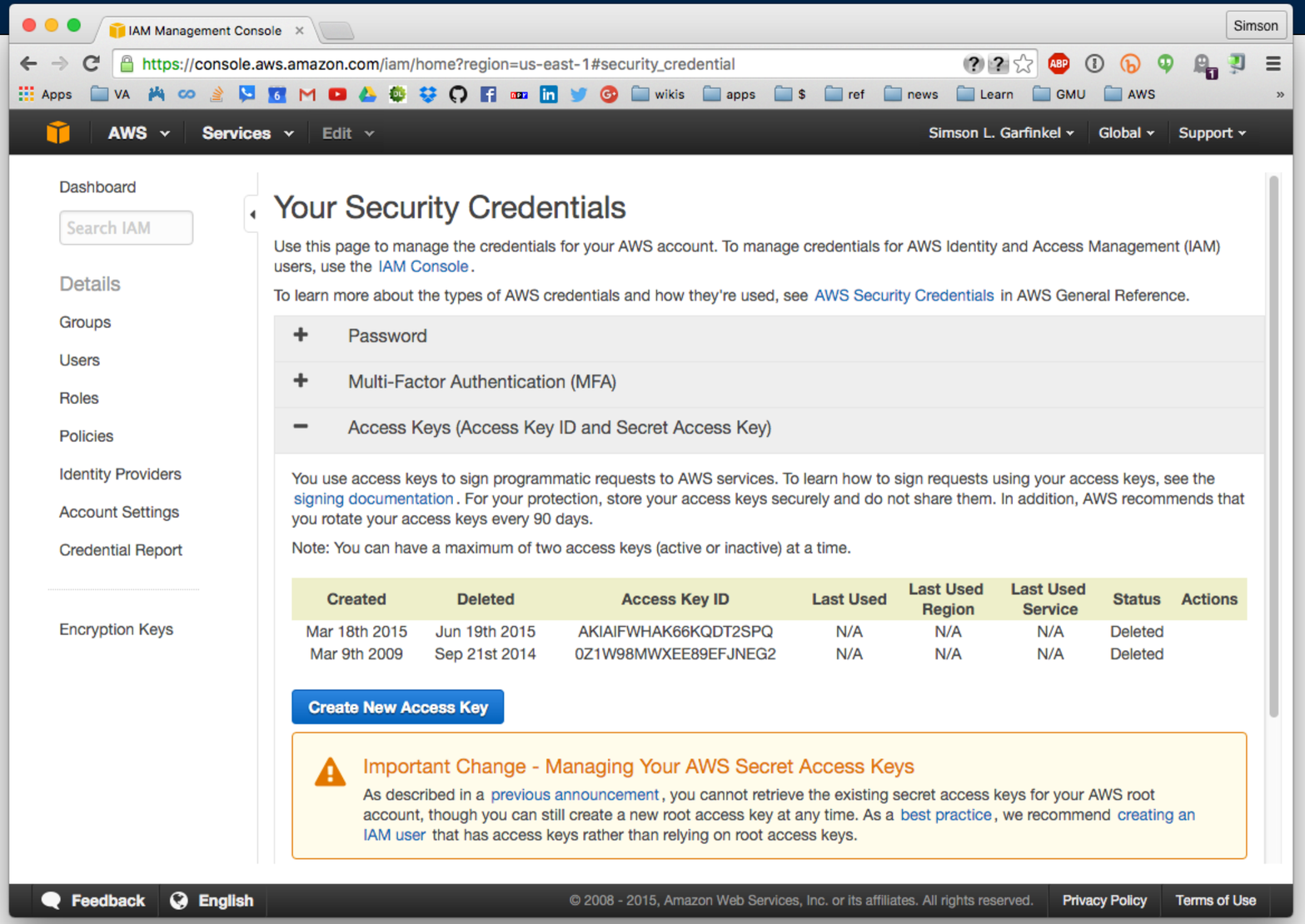
Two-factor authentication

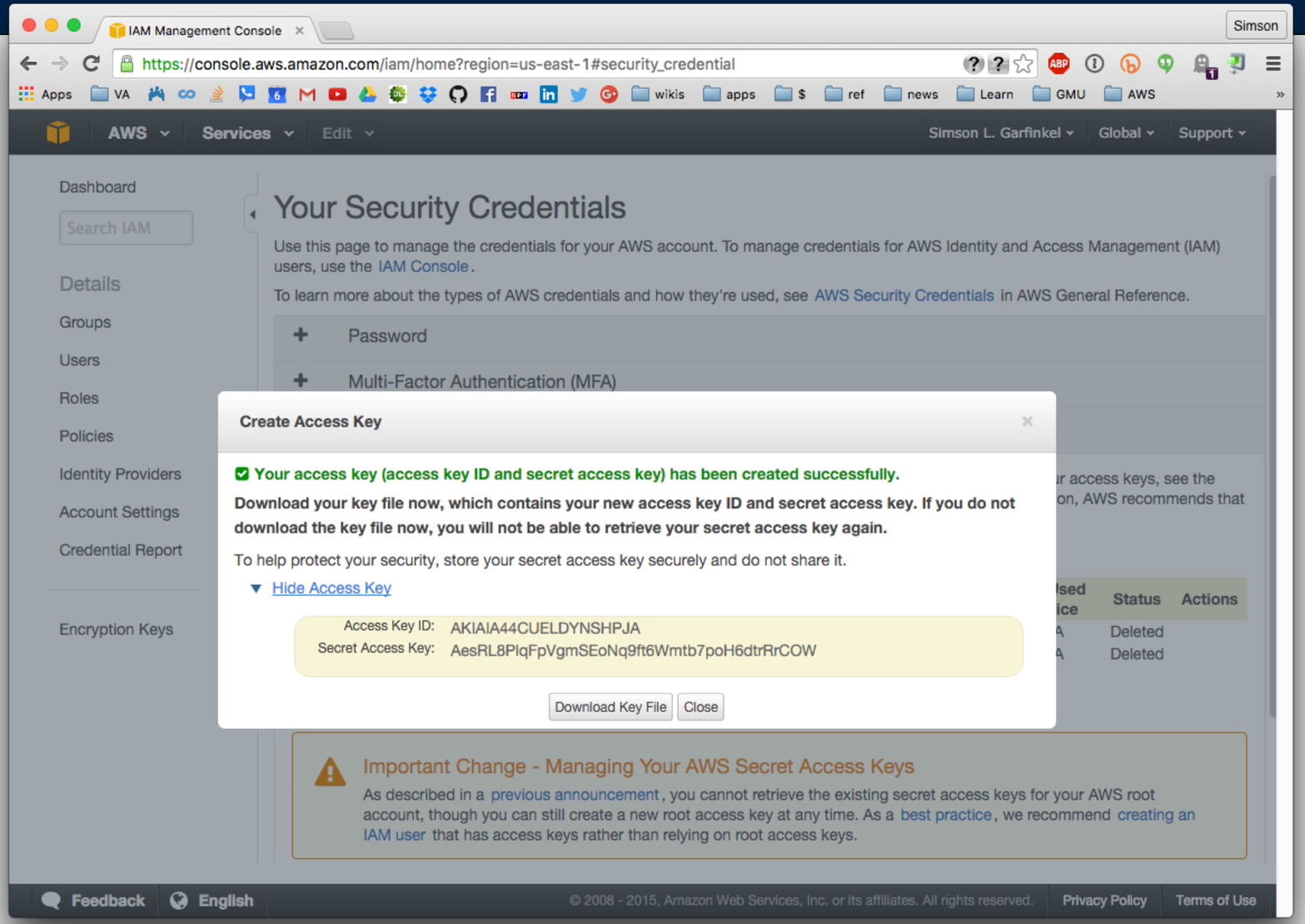
Account vs. Instance Credentials

AWS_KEY vs. AWS_SECRET_KEY


AWS Management Console
Simson
https://console.aws.amazon.com/console/home?region=us-east-1
Apps VA wikis apps $ ref news Learn GMU AWS
AWS Services Edit Simson L. Garfinkel N. Virginia Support
My Account
Billing & Cost Management
Security Credentials
Sign Out
Amazon Web Services
Compute
EC2 Virtual Servers in the Cloud
EC2 Container Service Run and Manage Docker Containers
Elastic Beanstalk Run and Manage Web Apps
Lambda Run Code in Response to Events
Storage & Content Delivery
S3 Scalable Storage in the Cloud
CloudFront Global Content Delivery Network
Elastic File System PREVIEW Fully Managed File System for EC2
Glacier Archive Storage in the Cloud
Import/Export Snowball Large Scale Data Transport
Storage Gateway Hybrid Storage Integration
Database
RDS Managed Relational Database Service
DynamoDB Managed NoSQL Database
ElastiCache In-Memory Cache
Redshift Fast, Simple, Cost-Effective Data Warehousing
Developer Tools
CodeCommit Store Code in Private Git Repositories
CodeDeploy Automate Code Deployments
CodePipeline Release Software using Continuous Delivery
Management Tools
CloudWatch Monitor Resources and Applications
CloudFormation Create and Manage Resources with Templates
CloudTrail Track User Activity and API Usage
Config Track Resource Inventory and Changes
OpsWorks Automate Operations with Chef
Service Catalog Create and Use Standardized Products
Trusted Advisor Optimize Performance and Security
Security & Identity
Identity & Access Management Manage User Access and Encryption Keys
Directory Service Host and Manage Active Directory
Inspector PREVIEW Analyze Application Security
WAF Filter Malicious Web Traffic
Internet of Things
AWS IoT BETA Connect Devices to the C
Mobile Services
Mobile Hub BETA Build, Test, and Monitor Mobile apps
Cognito User Identity and App Data Synchronization
Device Farm Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
Mobile Analytics Collect, View and Export App Analytics
SNS Push Notification Service
Application Services
API Gateway Build, Deploy and Manage APIs
AppStream Low Latency Application Streaming
CloudSearch Managed Search Service
Elastic Transcoder Easy-to-Use Scalable Media Transcoding
SES Email Sending and Receiving Service
SQS Message Queue Service
SWF Workflow Service for Coordinating Application Components
Enterprise Applications
e Groups Learn more
a collection of
onment in your
Create a Group
Tag Editor
Additional Resources
Getting Started
Read our documentation or view our training to learn more about AWS.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
AWS re:Invent Announcements
Explore the next generation of AWS cloud capabilities. See what's new
Service Health
All services operating normally.
https://console.aws.amazon.com/console/home?region=us-east-1#


IAM Management Console
Simson
https://console.aws.amazon.com/iam/home?region=us-east-1#security_credential
Apps VA wikis apps $ ref news Learn GMU AWS
AWS Services Edit Simson L. Garfinkel Global Support
Dashboard
Search IAM
Details
Groups
Users
Roles
Policies
Identity Providers
Account Settings
Credential Report
Encryption Keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
+ Password
+ Multi-Factor Authentication (MFA)
+ Access Keys (Access Key ID and Secret Access Key)
+ CloudFront Key Pairs
+ X.509 Certificates
+ Account Identifiers
Feedback English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy Terms of Use


IAM Management Console
Simson
https://console.aws.amazon.com/iam/home?region=us-east-1#security_credential
Apps VA wikis apps $ ref news Learn GMU AWS
AWS Services Edit Simson L. Garfinkel Global Support
Dashboard
Search IAM
Details
Groups
Users
Roles
Policies
Identity Providers
Account Settings
Credential Report
Encryption Keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
+ Password
+ Multi-Factor Authentication (MFA)
− Access Keys (Access Key ID and Secret Access Key)
You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.
Note: You can have a maximum of two access keys (active or inactive) at a time.
Created Deleted Access Key ID Last Used Last Used Region Last Used Service Status Actions
Mar 18th 2015 Jun 19th 2015 AKIAIFWHAK66KQDT2SPQ N/A N/A N/A Deleted
Mar 9th 2009 Sep 21st 2014 0Z1W98MWXEE89EFJNEG2 N/A N/A N/A Deleted
Create New Access Key
Important Change - Managing Your AWS Secret Access Keys
As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.
Feedback English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy Terms of Use


IAM Management Console
Simson
https://console.aws.amazon.com/iam/home?region=us-east-1#security_credential
Apps VA wikis apps $ ref news Learn GMU AWS
AWS Services Edit Simson L. Garfinkel Global Support
Dashboard
Search IAM
Details
Groups
Users
Roles
Policies
Identity Providers
Account Settings
Credential Report
Encryption Keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
Password
Multi-Factor Authentication (MFA)
Create Access Key
Your access key (access key ID and secret access key) has been created successfully.
Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.
To help protect your security, store your secret access key securely and do not share it.
Hide Access Key
Access Key ID: AKIAIA44CUELDYNSHPJA
Secret Access Key: AesRL8PlqFpVgmSEoNq9ft6Wmtb7poH6dtrRrCOW
Download Key File
Close
Status
Actions
Deleted
Deleted
Important Change - Managing Your AWS Secret Access Keys
As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

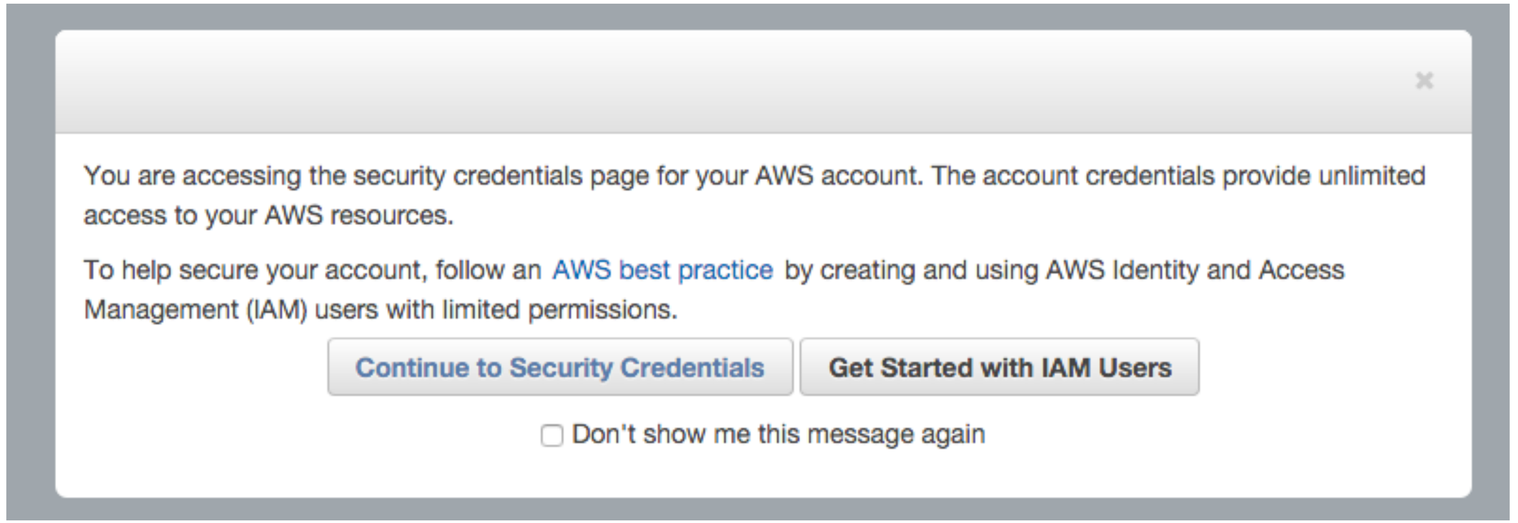
Best practices

Access keys:

- Don’t store in code. Store in an AWS credential file, environment variables, ~/.boto file, etc.

IAM users:

- Have their own username & password
- Can have authentication “burned in” to a EC2 instance
    - http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html



Security Credentials

Don’t use your AWS account access keys; create an IAM account.

Each IAM account has its own:

- Username, password, etc.
- Groups and Policies (what it can do)
- Access Keys (for API control.)
- It’s own multi-factor authentication.

Why use IAM accounts?

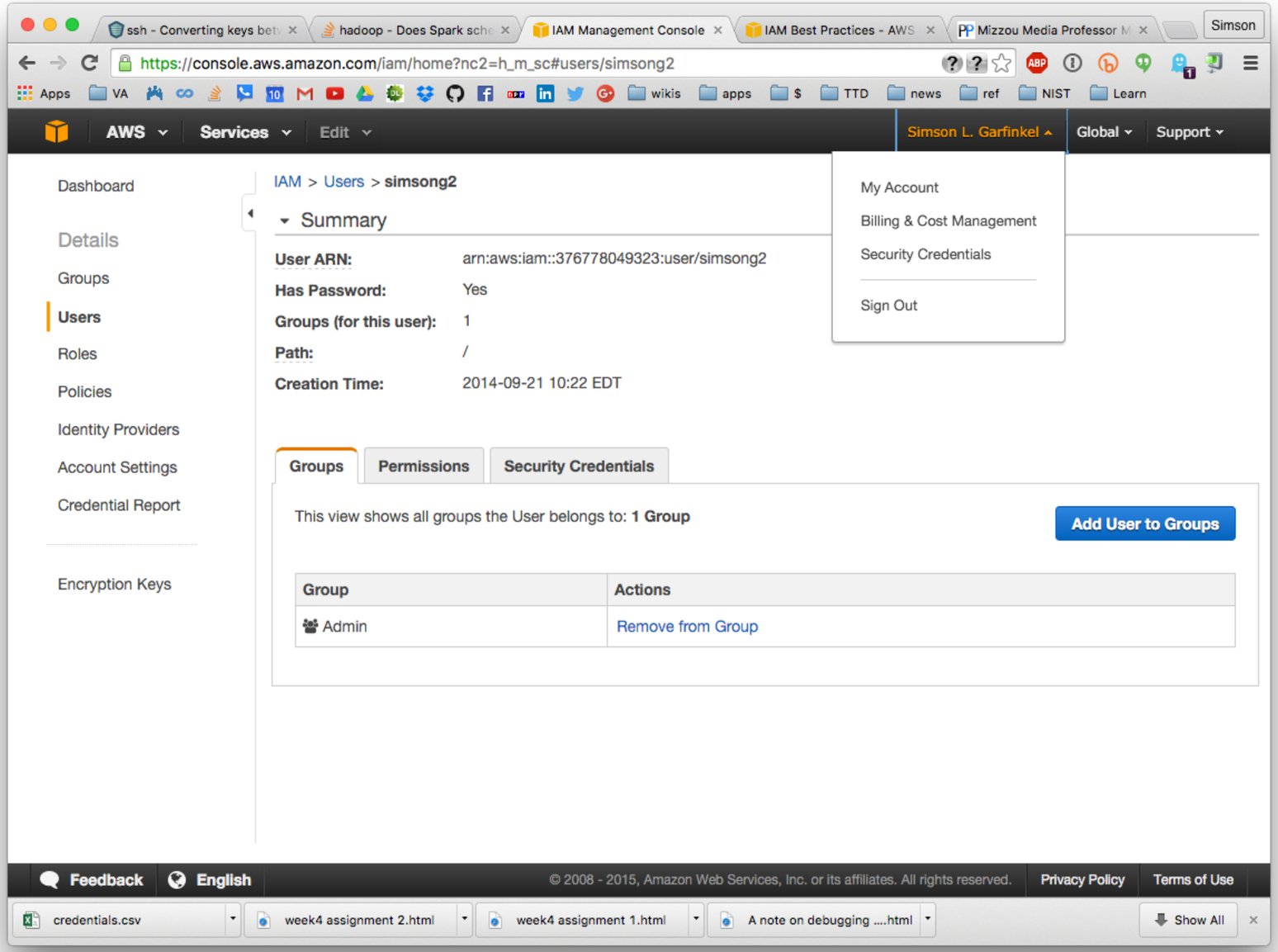
- Prevents a single user from wiping your entire cluster, data, etc.
- Allows you to create users that can only do a few things, without changing configurations:
  - *View data*
  - *Update data*

When you use IAM, you have a different name and login

My accounts:

- Account name: simsong
- Account sign in: https://simsong.signin.aws.amazon.com/console

By default, new users are in the "Admin" group



Demo:

Running Amazon EMR

Development Cycle: Move from concept to execution

Don't waste money! Prototype in a small VM

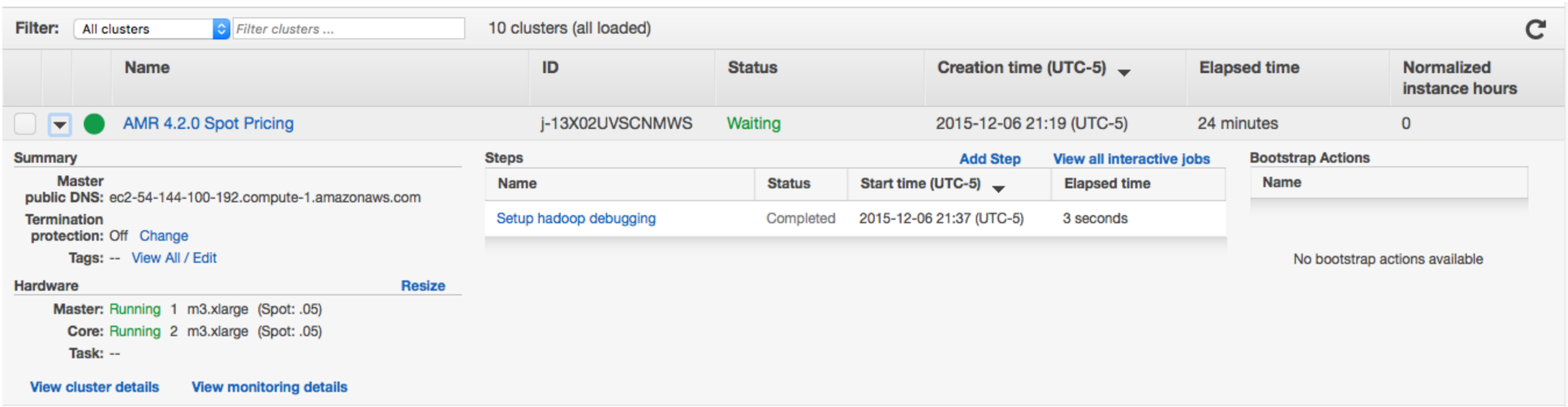
Approach:

- Test "locally" with mrjob without Hadoop (-r inline; -r local)
  - —*Cloudera VM*
  - —*Amazon instance*
  - —*Note: You cannot access hdfs:// or s3n:// with -r inline or -r local.*
- Spin up a small cluster (1 controller) and test with a subset of your data (-r hadoop)
- Add workers and/or data storage to your network, and run at scale.

Ways to improve development experience:

- Keep data in S3 — it's persistent, and frequently faster.
- Keep code in git.

Launch an EMR!



Access to the cluster:

```
ssh ec2-user@54.227.100.192
ssh hadoop@54.227.100.192
```

administrative tasks
running EMR

1. Copy over BCES_FY2014_clean.csv and mrjob_salary_max.py
2. Install mrjob

```
$ scp BCES_FY2014_clean.csv mrjob_salary_max.py hadoop@54.144.100.192:.
BCES_FY2014_clean.csv                         100% 1730KB   1.7MB/s   00:00
mrjob_salary_max.py                           100%  806     0.8KB/s   00:00
$ ssh hadoop@ec2-54-144-100-192.compute-1.amazonaws.com
X11 forwarding request failed on channel 0
Last login: Mon Dec  7 02:49:58 2015 from c-69-143-188-132.hsd1.va.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|
...
$ sudo pip install mrjob
...
[hadoop@ip-10-225-173-92 ~]$
$ export HADOOP_HOME=/usr/lib/hadoop
```

— *Test locally:*

```
$  python mrjob_salary_max.py -r local BCES_FY2014_clean.csv
```

— *Run on Hadoop:*

```
$ export HADOOP_HOME=/usr/lib/hadoop
$ python mrjob_salary_max.py -r hadoop BCES_FY2014_clean.csv
no configs found; falling back on auto-configuration
no configs found; falling back on auto-configuration
creating tmp directory /tmp/mrjob_salary_max.hadoop.20151207.025930.123916
writing wrapper script to /tmp/mrjob_salary_max.hadoop.20151207.025930.123916/setup-wrapper.sh
Using Hadoop version 2.6.0
Copying local files into hdfs:///user/hadoop/tmp/mrjob/mrjob_salary_max.hadoop.20151207.025930.123916/files/
```

```
HADOOP: Running job: job_1449455698873_0002
HADOOP: Job job_1449455698873_0002 running in uber mode : false
HADOOP:  map 0% reduce 0%
HADOOP:  map 6% reduce 0%
HADOOP:  map 13% reduce 0%
HADOOP:  map 19% reduce 0%
HADOOP:  map 56% reduce 0%
HADOOP:  map 63% reduce 0%
HADOOP:  map 69% reduce 0%
HADOOP:  map 75% reduce 0%
HADOOP:  map 81% reduce 0%
HADOOP:  map 88% reduce 0%
HADOOP:  map 100% reduce 0%
HADOOP:  map 100% reduce 14%
HADOOP:  map 100% reduce 57%
HADOOP:  map 100% reduce 71%
HADOOP:  map 100% reduce 100%
HADOOP: Job job_1449455698873_0002 completed successfully
HADOOP: Counters: 52
HADOOP:               File System Counters
HADOOP:                       FILE: Number of bytes read=19260
HADOOP:                       FILE: Number of bytes written=2757032
HADOOP:                       FILE: Number of read operations=0
HADOOP:                       FILE: Number of large read operations=0
HADOOP:                       FILE: Number of write operations=0
HADOOP:                       HDFS: Number of bytes read=2076133
HADOOP:                       HDFS: Number of bytes written=2509
HADOOP:                       HDFS: Number of read operations=69
HADOOP:                       HDFS: Number of large read operations=0
HADOOP:                       HDFS: Number of write operations=14
HADOOP:               Job Counters
HADOOP:                       Killed map tasks=1
HADOOP:                       Launched map tasks=17
HADOOP:                       Launched reduce tasks=7
HADOOP:                       Data-local map tasks=9
HADOOP:                       Rack-local map tasks=8
HADOOP:                       Total time spent by all maps in occupied slots (ms)=1317342
HADOOP:                       Total time spent by all reduces in occupied slots (ms)=571572
HADOOP:                       Total time spent by all map tasks (ms)=219557
HADOOP:                       Total time spent by all reduce tasks (ms)=47631
HADOOP:                       Total vcore-seconds taken by all map tasks=219557
HADOOP:                       Total vcore-seconds taken by all reduce tasks=47631
HADOOP:                       Total megabyte-seconds taken by all map tasks=316162080
HADOOP:                       Total megabyte-seconds taken by all reduce tasks=137177280
```

```
HADOOP:           Map-Reduce Framework
HADOOP:                     Map input records=18981
HADOOP:                     Map output records=34739
HADOOP:                     Map output bytes=4233496
HADOOP:                     Map output materialized bytes=29622
HADOOP:                     Input split bytes=3056
HADOOP:                     Combine input records=34739
HADOOP:                     Combine output records=320
HADOOP:                     Reduce input groups=2
HADOOP:                     Reduce shuffle bytes=29622
HADOOP:                     Reduce input records=320
HADOOP:                     Reduce output records=20
HADOOP:                     Spilled Records=640
HADOOP:                     Shuffled Maps =112
HADOOP:                     Failed Shuffles=0
HADOOP:                     Merged Map outputs=112
HADOOP:                     GC time elapsed (ms)=3014
HADOOP:                     CPU time spent (ms)=31080
HADOOP:                     Physical memory (bytes) snapshot=8335257600
HADOOP:                     Virtual memory (bytes) snapshot=51093340160
HADOOP:                     Total committed heap usage (bytes)=10174857216
HADOOP:           Shuffle Errors
HADOOP:                     BAD_ID=0
HADOOP:                     CONNECTION=0
HADOOP:                     IO_ERROR=0
HADOOP:                     WRONG_LENGTH=0
HADOOP:                     WRONG_MAP=0
HADOOP:                     WRONG_REDUCE=0
HADOOP:           File Input Format Counters
HADOOP:                     Bytes Read=2073077
HADOOP:           File Output Format Counters
HADOOP:                     Bytes Written=2509
HADOOP:           warn
HADOOP:                     missing gross=3223
HADOOP: Output directory: hdfs:///user/hadoop/tmp/mrjob/mrjob_salary_max.hadoop.20151207.025930.123916/output
```

```
Counters from step 1:
  (no counters found)
Streaming final output from hdfs:///user/hadoop/tmp/mrjob/mrjob_salary_max.hadoop.20151207.025930.123916/output
"salary"        [238772.0, "\"Bernstein,Gregg L\",STATE'S ATTORNEY,A29001,States Attorneys Office ,01/03/2011,$238772.00,$238772.04"]
"salary"        [200000.0, "\"Charles,Ronnie E\",EXECUTIVE LEVEL III,A83001,HR-Human Resources ,07/05/2012,$200000.00,$185741.81"]
"salary"        [193800.0, "\"Batts,Anthony W\",EXECUTIVE LEVEL III,A99390,Police Department ,09/25/2012,$193800.00,$193653.69"]
"salary"        [190000.0, "\"Black,Harry E\",EXECUTIVE LEVEL III,A23001,FIN-Admin & Budgets ,01/30/2012,$190000.00,$188328.50"]
"salary"        [187200.0, "\"Swift,Michael\",CONTRACT SERV SPEC II,A02003,City Council ,05/19/2008,$187200.00,$3510.00"]
"salary"        [172000.0, "\"Parthemos,Kaliope\",EXECUTIVE LEVEL III,A01020,Mayor's Office ,12/26/2006,$172000.00,$154654.39"]
"salary"        [165000.0, "\"Ford,Niles R\",EXECUTIVE LEVEL III,A64006,Fire Department ,01/15/2014,$165000.00,$69807.64"]
"salary"        [163365.0, "\"Rawlings-Blake,Stephanie C\",MAYOR,A01001,Mayors Office ,12/07/1995,$163365.00,$161219.24"]
"salary"        [163200.0, "\"Nilson,George A\",CITY SOLICITOR,A30001,Law Department ,01/16/2007,$163200.00,$164332.32"]
"salary"        [163200.0, "\"Chow,Rudolph S\",DIRECTOR PUBLIC WORKS,A41101,DPW-Administration ,02/01/2011,$163200.00,$145513.79"]
"gross"         [238772.04000000001, "\"Bernstein,Gregg L\",STATE'S ATTORNEY,A29001,States Attorneys Office ,
01/03/2011,$238772.00,$238772.04"]
"gross"         [193653.69, "\"Batts,Anthony W\",EXECUTIVE LEVEL III,A99390,Police Department ,09/25/2012,$193800.00,$193653.69"]
"gross"         [188328.5, "\"Black,Harry E\",EXECUTIVE LEVEL III,A23001,FIN-Admin & Budgets ,01/30/2012,$190000.00,$188328.50"]
"gross"         [185741.81, "\"Charles,Ronnie E\",EXECUTIVE LEVEL III,A83001,HR-Human Resources ,07/05/2012,$200000.00,$185741.81"]
"gross"         [176141.32999999999, "\"Nalewajko Jr,Stephen C\",POLICE LIEUTENANT EID,A99264,Police Department ,
08/21/1981,$95087.00,$176141.33"]
"gross"         [173876.84, "\"Marcus Sr,Albert M\",POLICE OFFICER (EID),A99322,Police Department ,02/03/1975,$73012.00,$173876.84"]
"gross"         [166442.42000000001, "\"Stokes,Charline B\",Battalion Fire Chief EMS EMT-P,A64460,Fire Department ,
01/18/1988,$107307.00,$166442.42"]
"gross"         [165892.20999999999, "\"Harris Jr,William\",POLICE SERGEANT,A99309,Police Department ,
10/24/2000,$80612.00,$165892.21"]
"gross"         [165270.01000000001, "\"Makanjuola,Rafiu T\",POLICE OFFICER (EID),A99061,Police Department ,
07/30/1997,$67535.00,$165270.01"]
"gross"         [165108.5, "\"Cheelsman III,Charles H\",Battalion Fire Chief EMS EMT-P,A64460,Fire Department ,
12/08/1980,$107307.00,$165108.50"]
removing tmp directory /tmp/mrjob_salary_max.hadoop.20151207.025930.123916
deleting hdfs:///user/hadoop/tmp/mrjob/mrjob_salary_max.hadoop.20151207.025930.123916 from HDFS
```

It worked!

Hadoop's overhead is costly (here)

- Total time to run with -r local: 0m2s
- Total time to run with -r hadoop: 1m25s

Notice that mrjob:

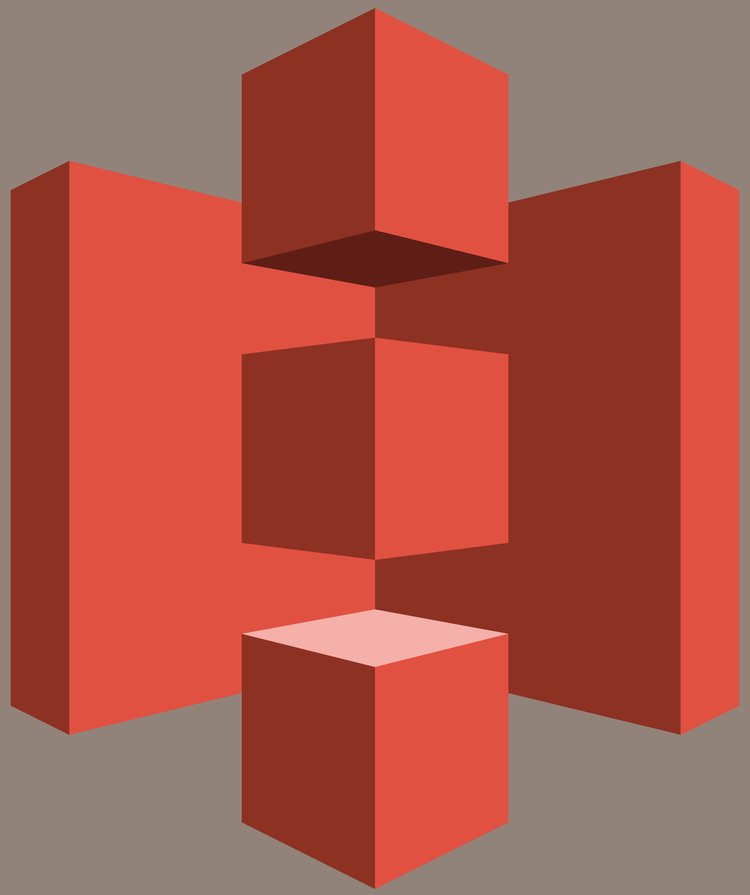
- copied data from local file system into HDFS
- ran hadoop
- copied data out from HDFS
- Deleted the temp files.

Run with data in HDFS:

```
$ hdfs dfs -mkdir /user/hadoop/in1/
$ hdfs dfs -put BCES_FY2014_clean.csv /user/hadoop/in1/
$ python mrjob_salary_max.py -r hadoop hdfs:///user/hadoop/in1/BCES_FY2014_clean.csv
```

Run with data in S3 (data copied from S3 back to HDFS):

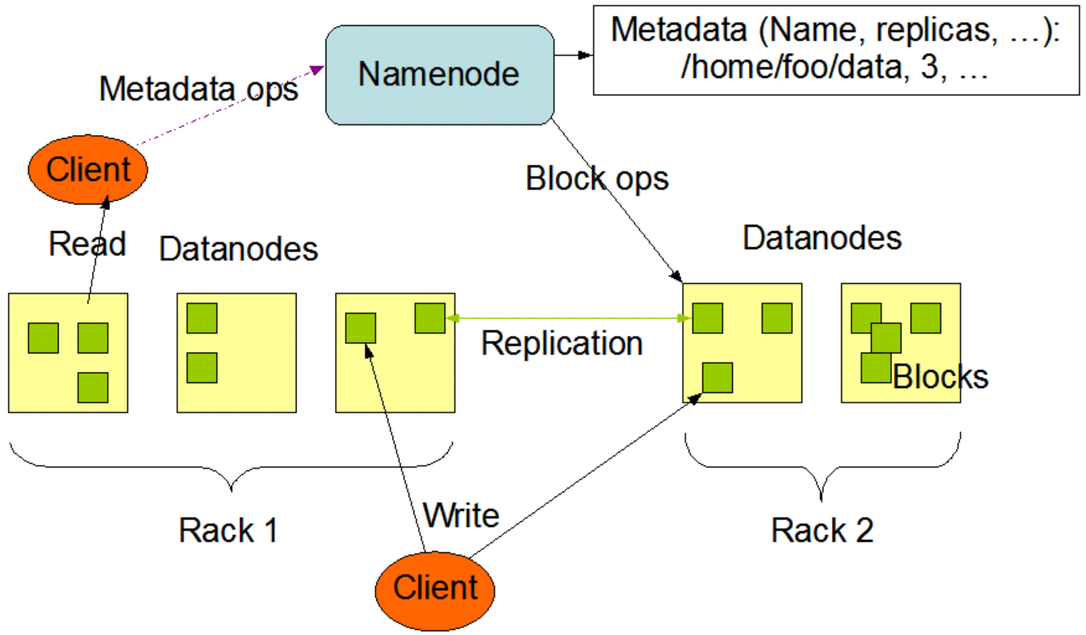
```
$ aws s3 cp BCES_FY2014_clean.csv s3://slgemr/in1/BCES_FY2014_clean.csv
$ python mrjob_salary_max.py -r hadoop s3://slgemr/in1/
```



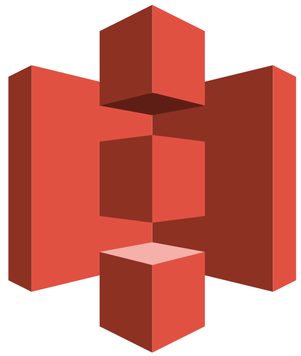
Amazon S3 — Deep Dive

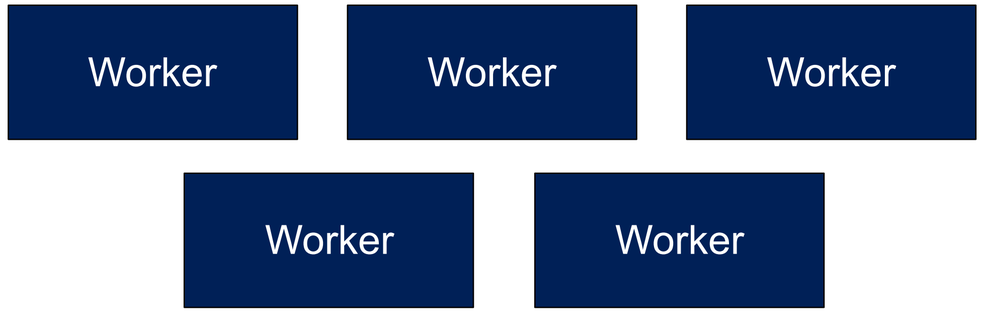
S3 vs. HDFS

HDFS:



Amazon S3:

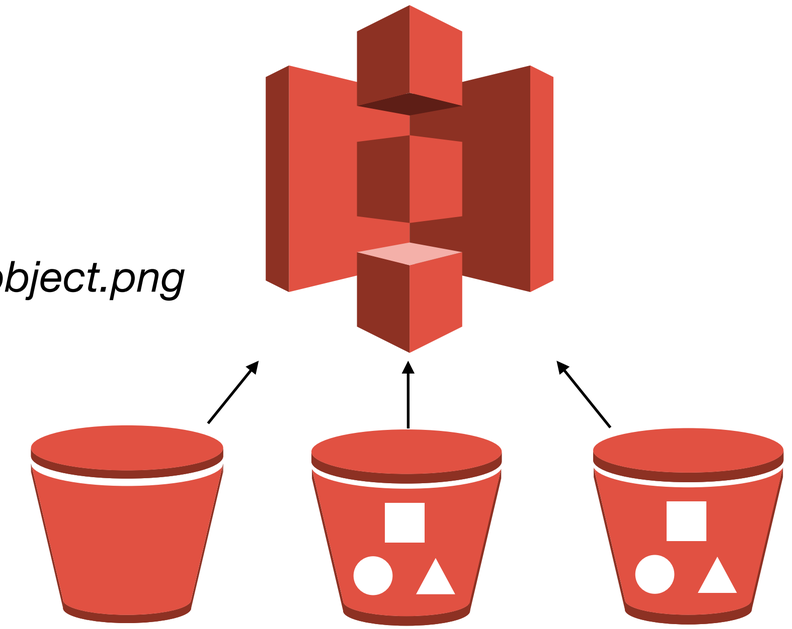




S3 arranges objects (files) into "buckets." Each user has one or more buckets

Each bucket:

- Has a unique name
- Has a unique URL
  - —*https://anly502.s3.amazonaws.com/object.png*

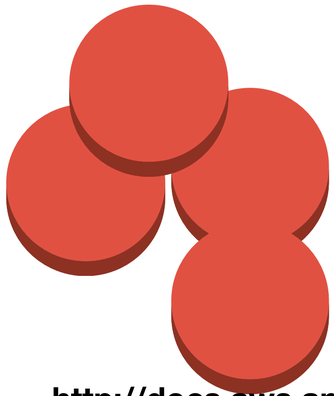


ANLY 502 Amazon S3 Account

S3 Buckets

Buckets can also:

- Set to a specific region
- Enable versioning
- Serve static HTML pages.
- Multiple consistency models.
- Reduced Redundancy Storage for lower cost.



Objects in an S3 bucket

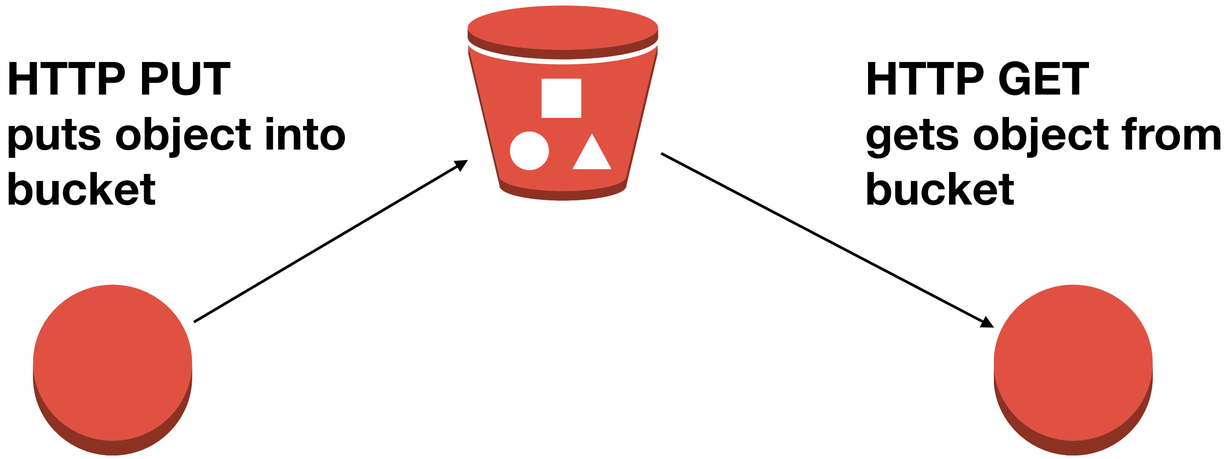
Objects:

- In buckets
- Identified by "keys" (e.g object.png, a/b/c/d/object.png)

http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

S3 access protocol: REST

REST is built on top of HTTP.



Many ways to access data on Amazon S3

HTTP / REST — Representational State Transfer

- Uses HTTP methods (with a bit of JSON)
- HTTP GET — Reads a resource without causing any side effects
- HTTP DELETE — Deletes a resources
- HTTP PUT (or POST) — Creates a new resources
- HTTP POST (or PUT) — Modify a resource's value

HTTP Hosting

- Different from REST
- Must be explicitly enabled

HTTP / SOAP — Simple Object Access Protocol

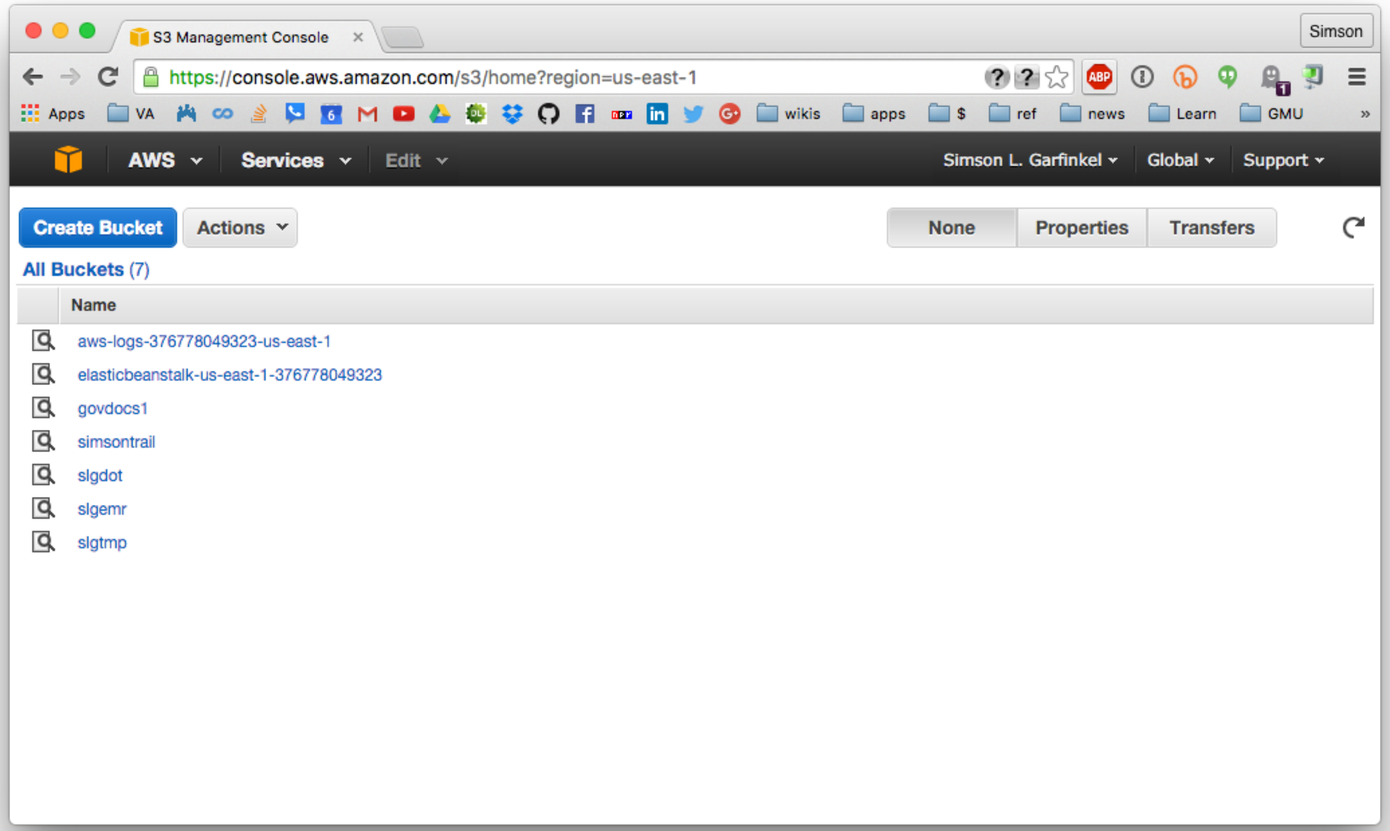
- Structure XML-based protocol
- Heavy weight; increasingly not used.

BitTorrent

- S3 can host a "tracker" and "seeds"
- Limited to objects 5GB in size

Buckets are controlled from web-API or CLI

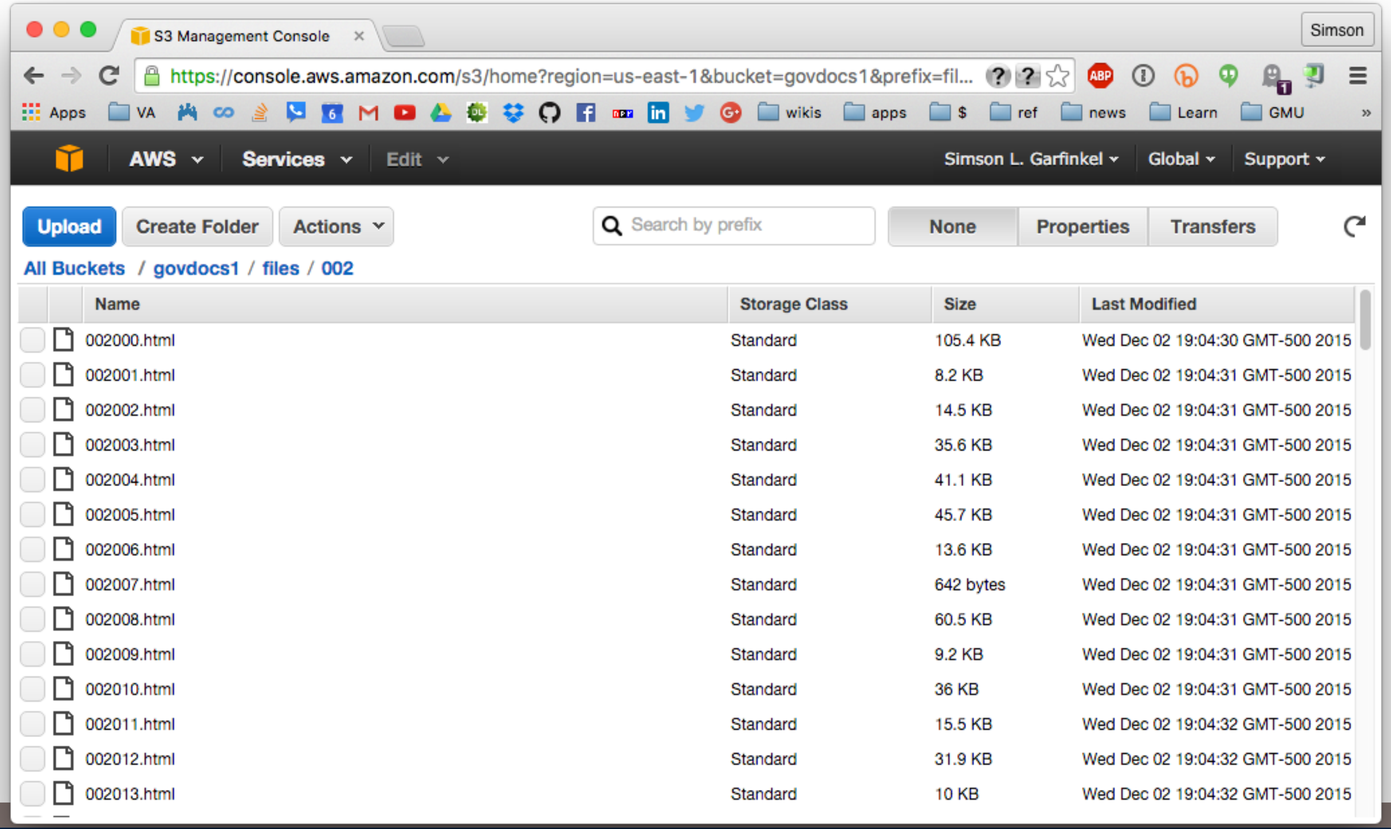
View all buckets



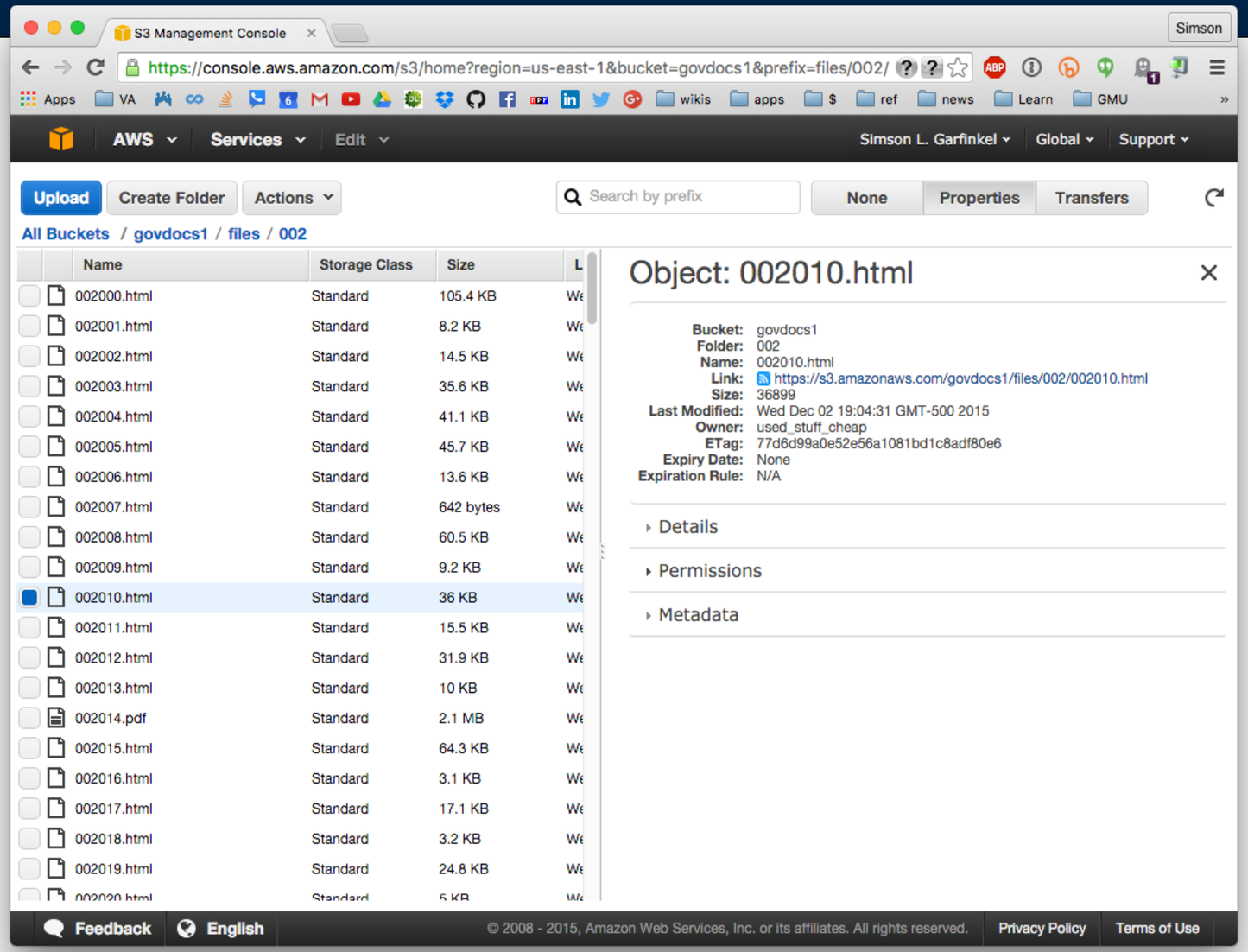
Drill down to a specific file in a bucket

Folders don't actually exist.

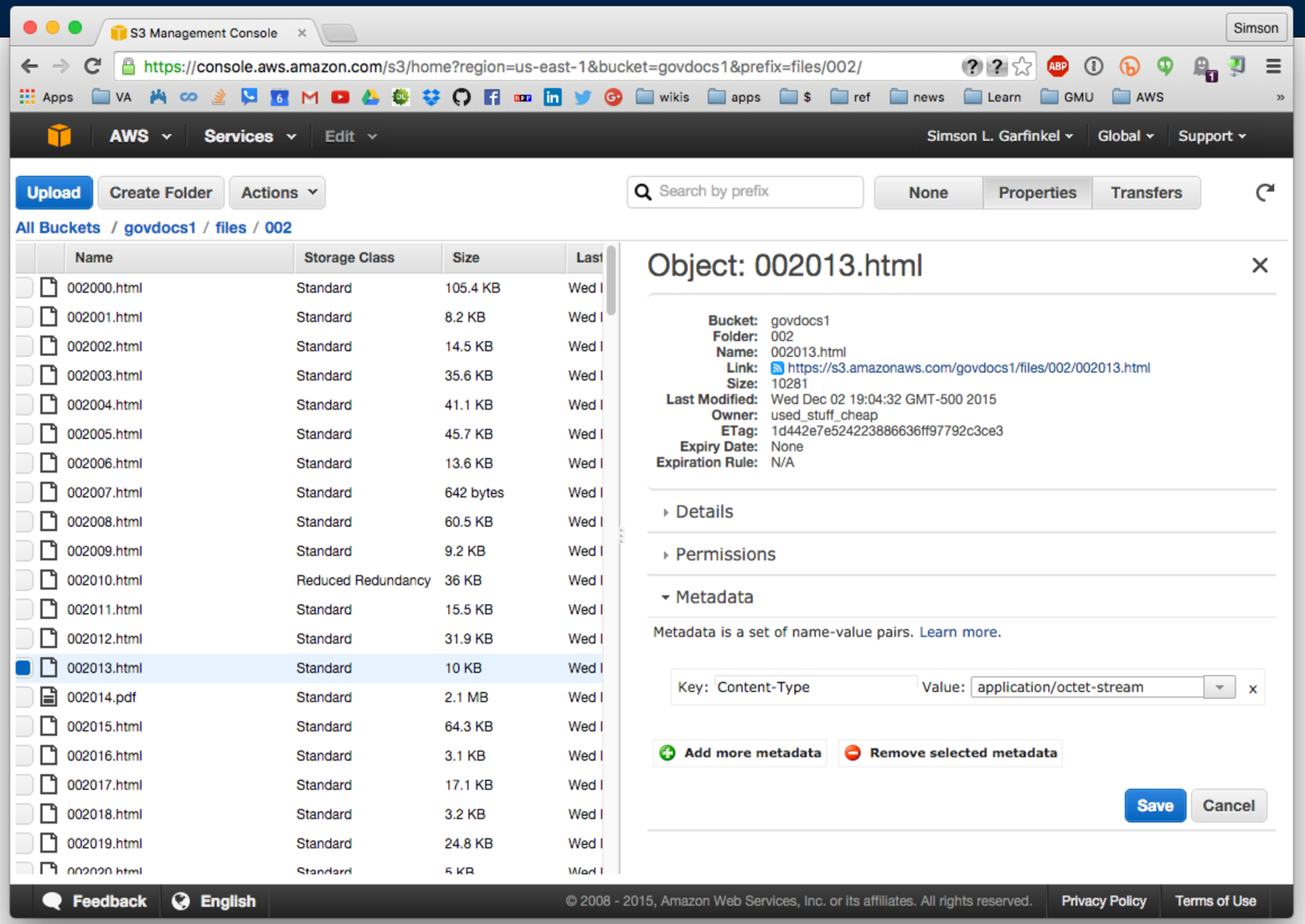
- "For the sake of organizational simplicity, the Amazon S3 console supports the folder concept as a means of grouping objects. Amazon S3 does this by using key name prefixes for objects."
  - http://docs.aws.amazon.com/AmazonS3/latest/UG/FolderOperations.html
- Remember: govdocs1/files/002 is not a directory — it's a prefix



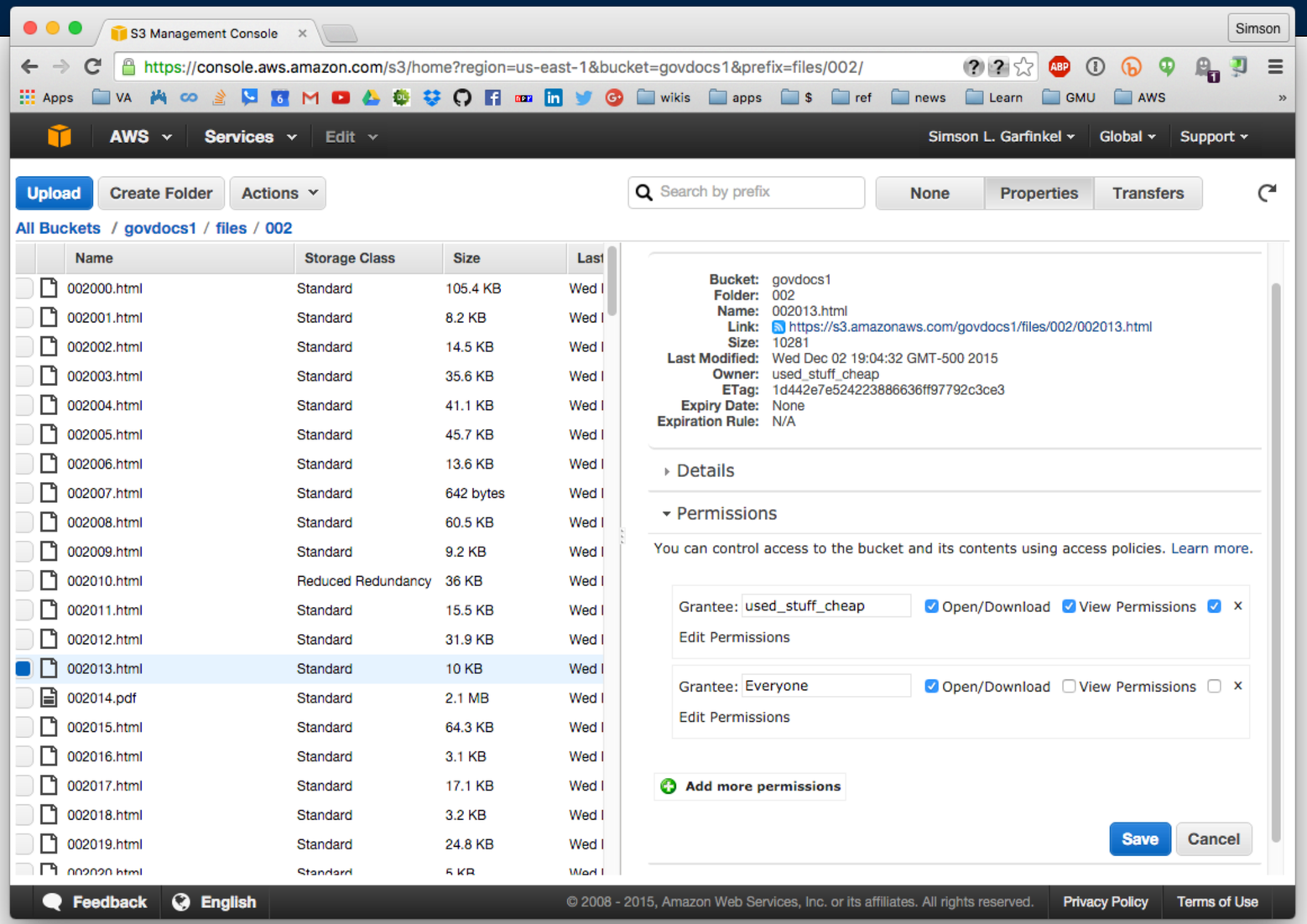
Each object has properties



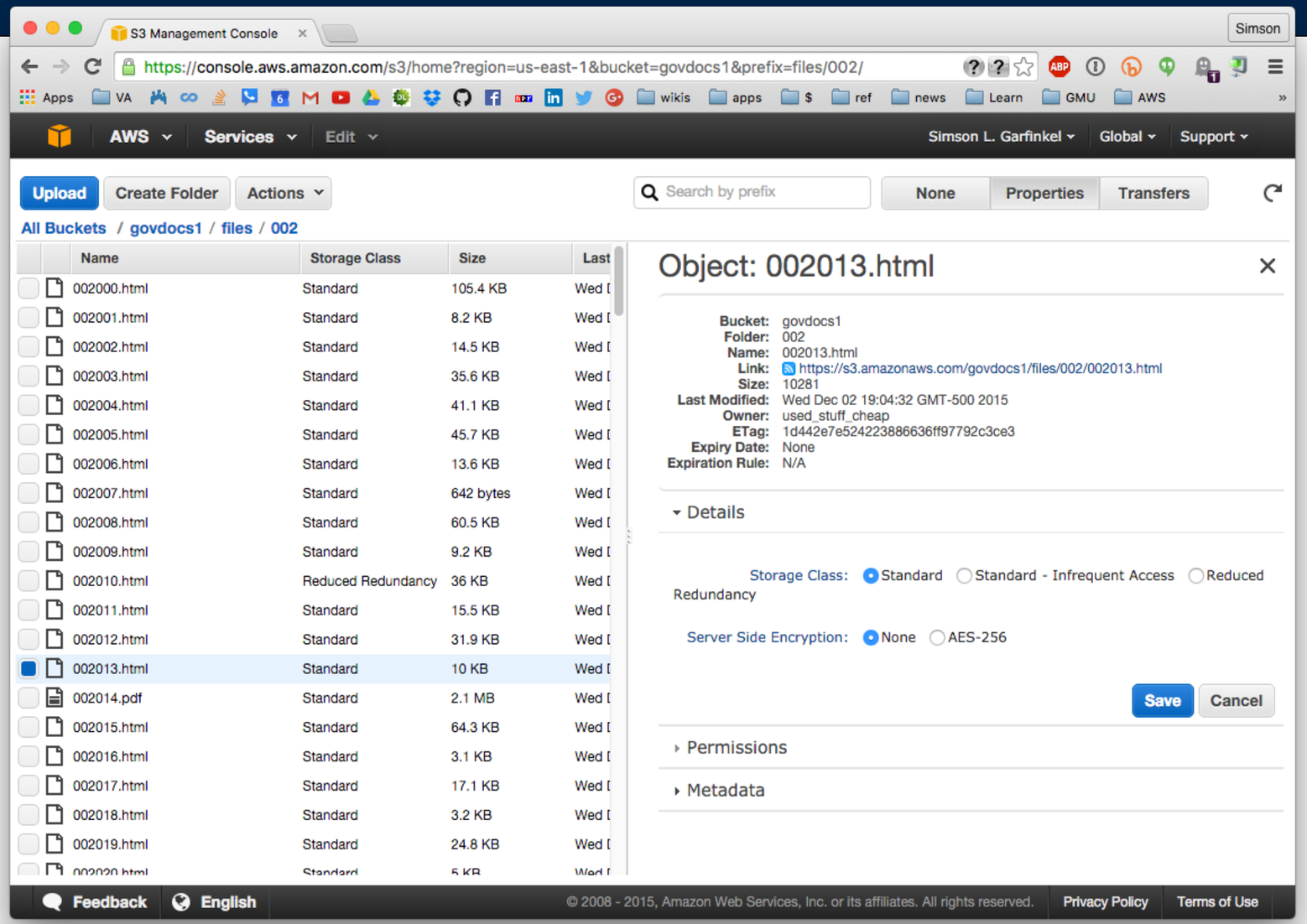
Each object has optional metadata



Permissions can be set per-bucket



Storage class can be set per-object or per-bucket



S3 Pricing — You pay for what you use (as of 2015-12-15)

AWS Free Usage Tier:

- 5GB of S3 Storage, 20,000 GET requests, 2,000 PUT requests, 15GB of data transfer free EACH MONTH for 1 year.

	Standard Storage	Standard - Infrequent Access Storage †	Reduced Redundancy	Glacier Storage
First 1 TB / month	$0.0300 per GB	$0.0125 per GB	$0.0240 per GB	$0.007 per GB
Next 49 TB / month	$0.0295 per GB	$0.0125 per GB	$0.0236 per GB	$0.007 per GB
Next 450 TB / month	$0.0290 per GB	$0.0125 per GB	$0.0228 per GB	$0.007 per GB
Next 500 TB / month	$0.0285 per GB	$0.0125 per GB	$0.220 per GB	$0.007 per GB
PUT, COPY, POST, LIST	$0.05 per 10,000	$0.10 per 10,000	$0.05 per 10,000	$0.50 per 10,000
GET requests	$0.004 per 10,000	$0.01 per 10,000	$0.004 per 10,000	$0.50 per 10,000
DEL	Free	Free	Free	‡

- http://aws.amazon.com/s3/pricing/
- http://aws.amazon.com/s3/reduced-redundancy/

‡ charged for deleting objects less than 3 months old

Pricing: S3 vs. EBS (as of 2015-12-15)

Price for 1TB

	Standard Storage	Standard - Infrequent Access Storage †	Reduced Redundancy	Glacier Storage
First 1 TB / month	$0.0300 per GB	$0.0125 per GB	$0.0240 per GB	$0.007 per GB
Next 49 TB / month	$0.0295 per GB	$0.0125 per GB	$0.0236 per GB	$0.007 per GB
Next 450 TB / month	$0.0290 per GB	$0.0125 per GB	$0.0228 per GB	$0.007 per GB
Next 500 TB / month	$0.0285 per GB	$0.0125 per GB	$0.220 per GB	$0.007 per GB
PUT, COPY, POST, LIST	$0.05 per 10,000	$0.10 per 10,000	$0.05 per 10,000	$0.50 per 10,000
GET requests	$0.004 per 10,000	$0.01 per 10,000	$0.004 per 10,000	$0.50 per 10,000
DEL	Free	Free	Free	‡

- http://aws.amazon.com/s3/pricing/
- http://aws.amazon.com/s3/reduced-redundancy/

‡ charged for deleting objects less than 3 months old

Pricing — Data Transfer (as of 2015-12-15)

Data Transfer IN To Amazon S3	
All data transfer in	$0.000 per GB
Data Transfer OUT From Amazon S3 To	
Amazon EC2 in the Northern Virginia Region	$0.000 per GB
Another AWS Region	$0.020 per GB
Amazon CloudFront	$0.000 per GB
Data Transfer OUT From Amazon S3 To Internet	
First 1 GB / month	$0.000 per GB
Up to 10 TB / month	$0.090 per GB
Next 40 TB / month	$0.085 per GB
Next 100 TB / month	$0.070 per GB
Next 350 TB / month	$0.050 per GB
Next 524 TB / month	Contact Us
Next 4 PB / month	Contact Us
Greater than 5 PB / month	Contact Us

Pricing S3: What is the cost of storing and accessing GOVDOCS1?

Corpus size: 1 million files, 500GB (average 0.5MB per file)

Storage: $0.03/GB x 500GB = $15/month

Access: $0.004/10,000 GETs x 1M GETs = $0.40

Upload files to S3 with Python: use “boto”

Simple program to upload the file ‘000.zip’
to s3://simsong/govdocs1/zipfiles/000.zip:

```
#!/usr/bin/env python
# https://aws.amazon.com/articles/Python/3998

import boto
s3 = boto.connect_s3()
bucket = s3.get_bucket('govdocs1')
key = bucket.new_key('zipfiles/z1')
key.set_contents_from_filename(‘000.zip')
key.set_acl(‘public-read')
```

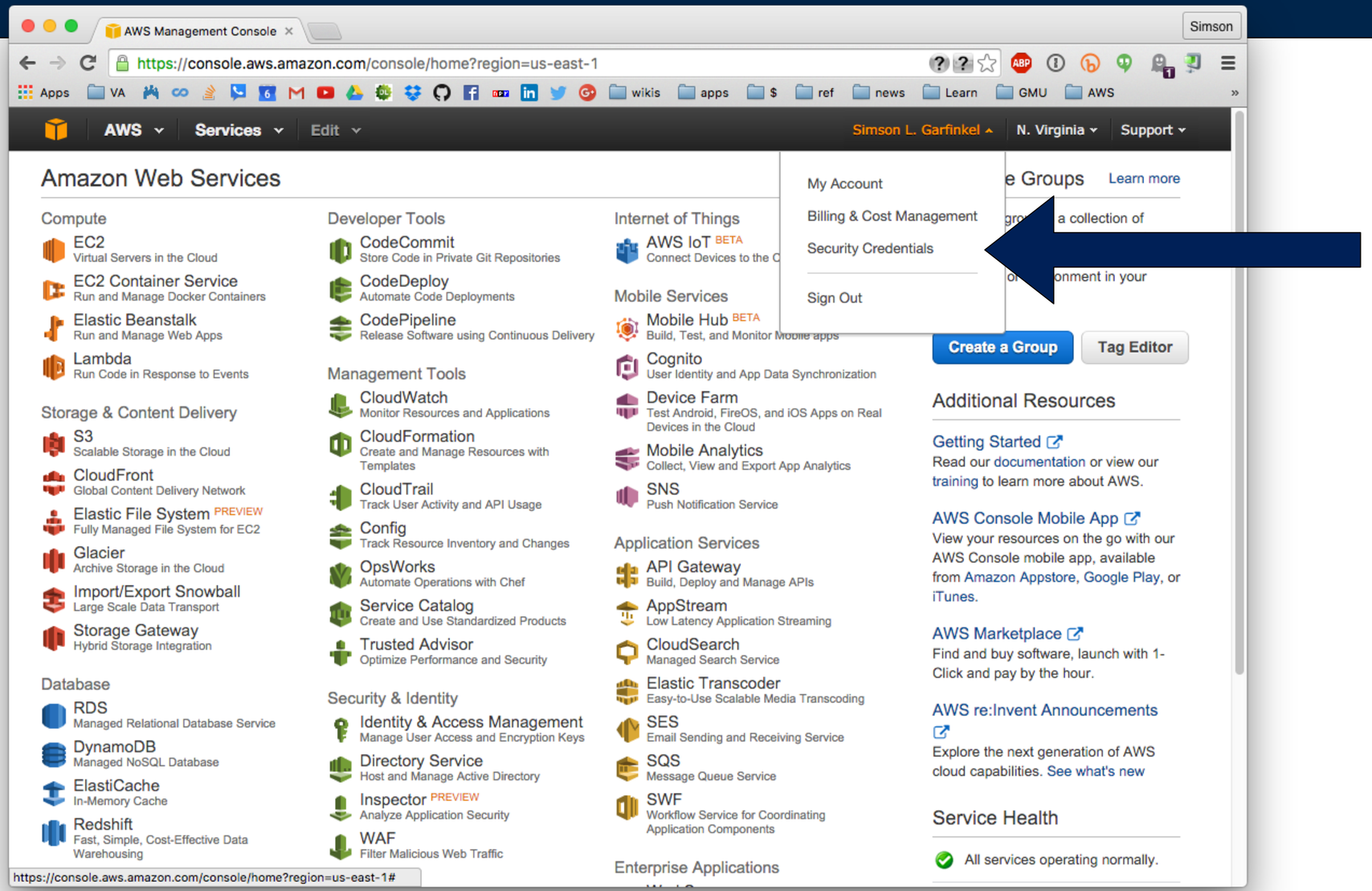
This program doesn’t work as-is….

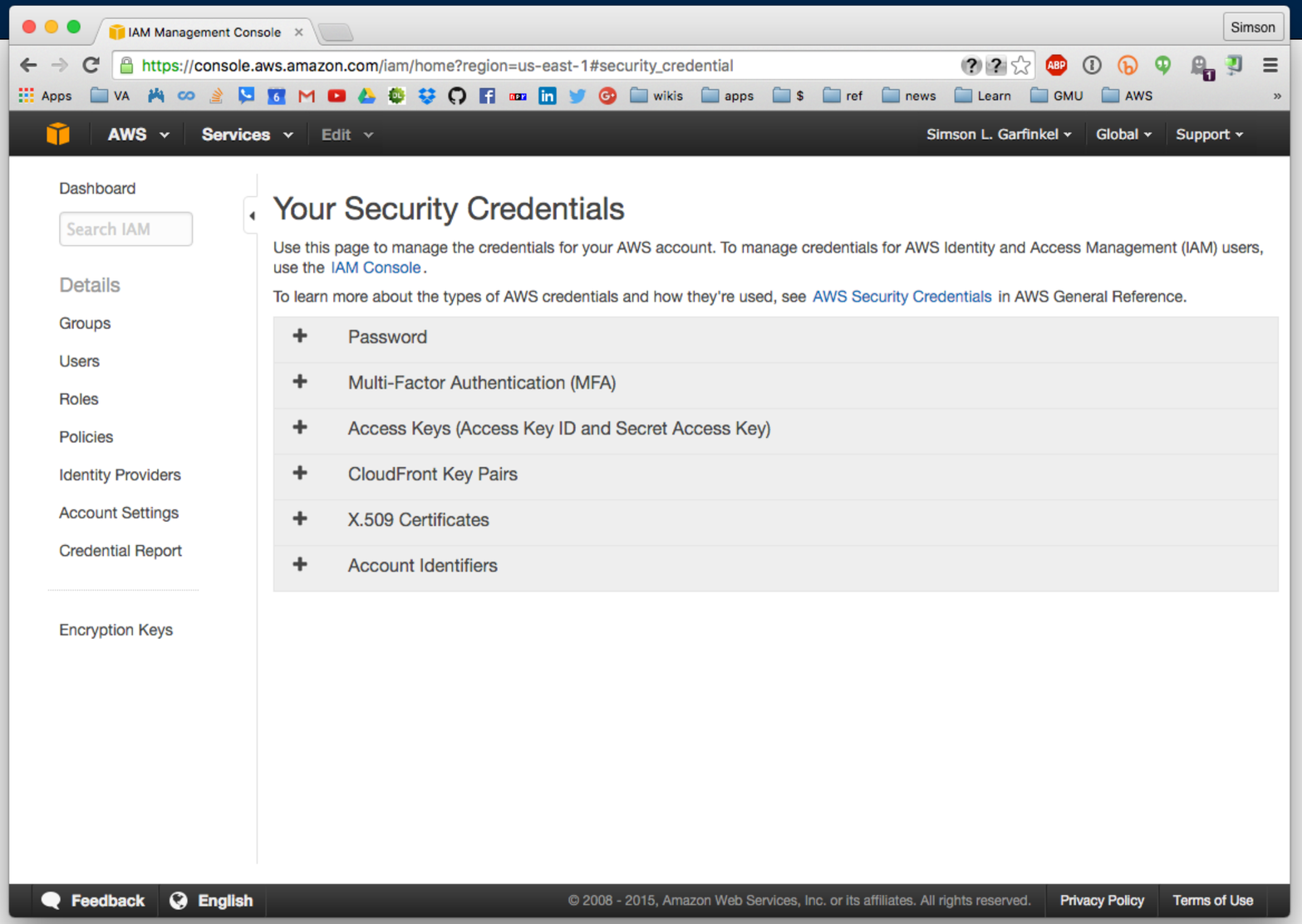
The uploader doesn't have permission to upload to S3.

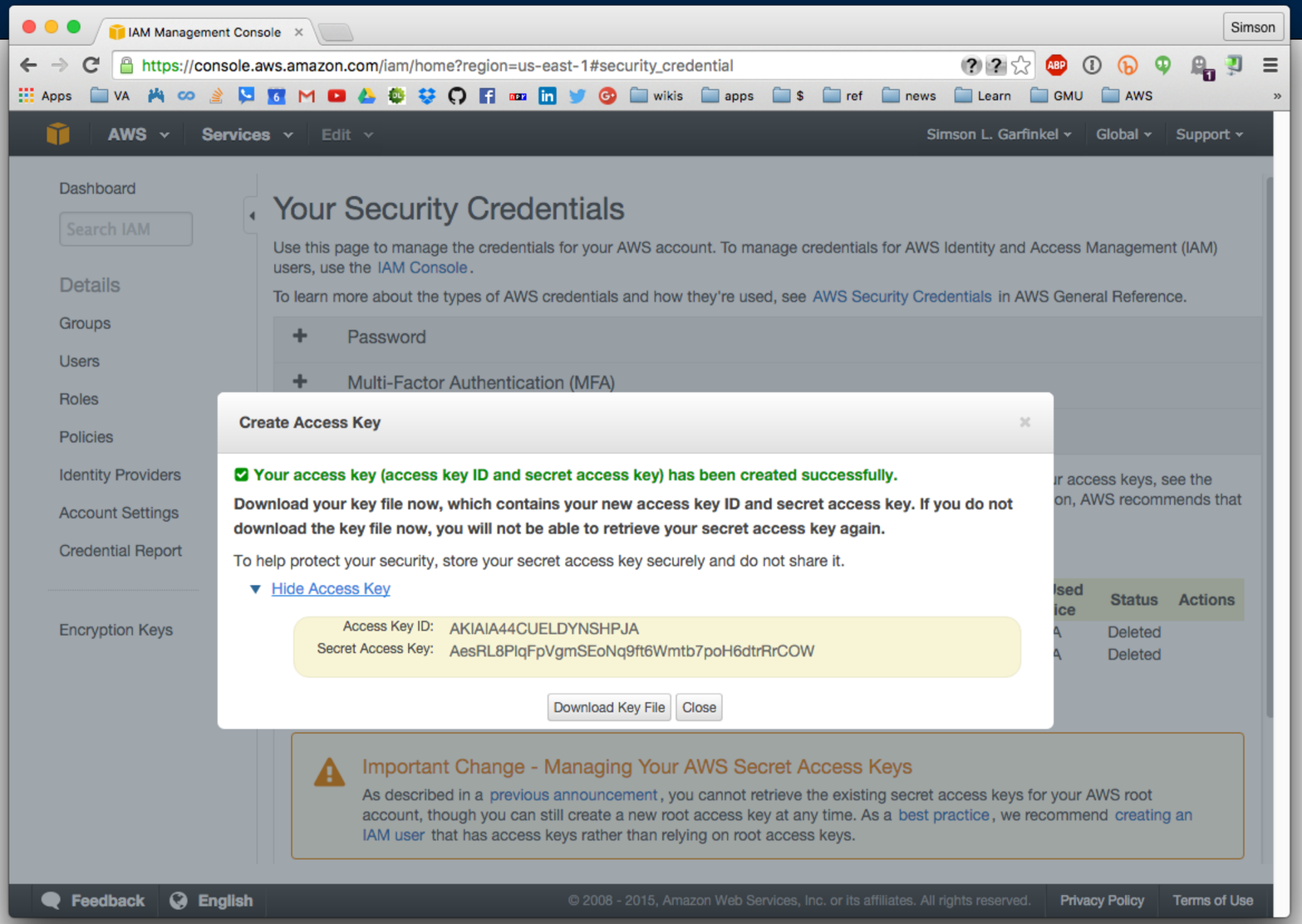
```
$ python uploader.py
Traceback (most recent call last):
  File "uploader.py", line 8, in <module>
    key.set_contents_from_filename('uploader.py')
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1362, in set_contents_from_filename
    encrypt_key=encrypt_key)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1293, in set_contents_from_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 750, in send_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 951, in _send_file_internal
    query_args=query_args
  File "/usr/lib/python2.7/dist-packages/boto/s3/connection.py", line 664, in make_request
    retry_handler=retry_handler
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1071, in make_request
    retry_handler=retry_handler)
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 940, in _mexe
    request.body, request.headers)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 884, in sender
    response.status, response.reason, body)
boto.exception.S3ResponseError: S3ResponseError: 403 Forbidden
<?xml version="1.0" encoding="UTF-8"?>
<Error><Code>AccessDenied</Code><Message>Access Denied</Message><RequestId>3C2C61650BFDDC0D</
RequestId><HostId>u7DYAYhQVUiaygmnScU0cmozaU8kRofETXoiH00yLC/8jYqcS4aNSfWRJaSWDu0GeKRFyzizQ28=</HostId></Error>
$
```

Boto requires AWS authentication

Create/get your security credentials




IAM Management Console
Simson
https://console.aws.amazon.com/iam/home?region=us-east-1#security_credential
Apps VA wikis apps $ ref news Learn GMU AWS
AWS Services Edit Simson L. Garfinkel Global Support
Dashboard
Search IAM
Details
Groups
Users
Roles
Policies
Identity Providers
Account Settings
Credential Report
Encryption Keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
+ Password
+ Multi-Factor Authentication (MFA)
+ Access Keys (Access Key ID and Secret Access Key)
+ CloudFront Key Pairs
+ X.509 Certificates
+ Account Identifiers
Feedback English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy Terms of Use


IAM Management Console
Simson
https://console.aws.amazon.com/iam/home?region=us-east-1#security_credential
AWS
Services
Edit
Simson L. Garfinkel
Global
Support
Dashboard
Search IAM
Details
Groups
Users
Roles
Policies
Identity Providers
Account Settings
Credential Report
Encryption Keys
Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
Password
Multi-Factor Authentication (MFA)
Create Access Key
Your access key (access key ID and secret access key) has been created successfully.
Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.
To help protect your security, store your secret access key securely and do not share it.
Hide Access Key
Access Key ID: AKIAIA44CUELDYNSHPJA
Secret Access Key: AesRL8PlqFpVgmSEoNq9ft6Wmtb7poH6dtrRrCOW
Download Key File
Close
Status
Actions
Deleted
Deleted
Important Change - Managing Your AWS Secret Access Keys
As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

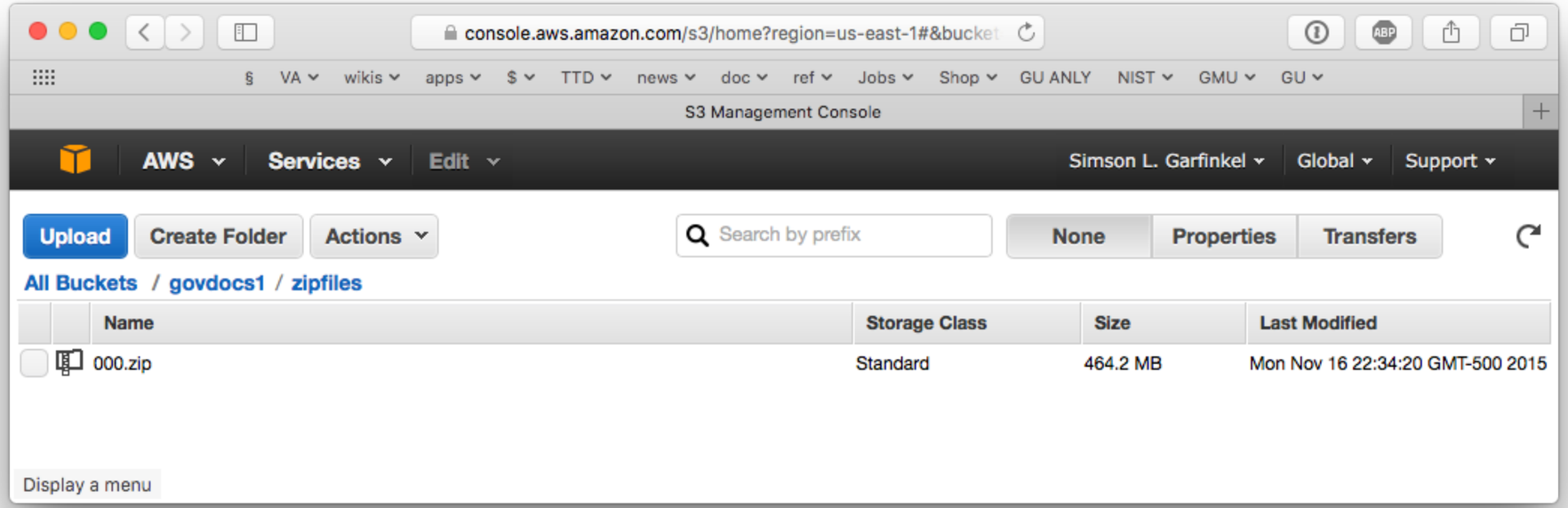
Boto requires AWS authentication
Store in the file ~/.boto

```
$ cat ~/.boto
[Credentials]
aws_access_key_id = AKIAJNMBJEXXYO6USPHD
aws_secret_access_key = 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
$
```

Now let's run it:

```
$ python uploader.py
$
```

And we can verify it's there:



S3 Authentication

Access to S3 is authenticated with two secret keys:

- AWS Access Key ID — 20 Character String
  - —*e.g. AKIAIOSFODNN7EXAMPLE*
- AWS Secret Key — 40-character string
  - —*e.g. wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

Keys can be created:

- For primary user account
- For IAM accounts
- For individual applications

Public key cryptography is not used!

- Credentials must be sent over SSL

Uploading to S3

Simple program to upload the file ‘000.zip’ to s3://simsong/govdocs1/zipfiles/000.zip:

```
#!/usr/bin/env python
# https://aws.amazon.com/articles/Python/3998

import boto
s3 = boto.connect_s3()
bucket = s3.get_bucket('govdocs1')
key = bucket.new_key('zipfiles/z1')
key.set_contents_from_filename(‘000.zip')
key.set_acl(‘public-read')
```

Permission errors on S3 can be obscure

This means you don't have permission to upload:

```
Uploading s3://govdocs1/400.zip
Traceback (most recent call last):
  File "uploader.py", line 75, in <module>
    results.append(copy(i))
  File "uploader.py", line 41, in copy
    key.set_contents_from_filename(fname)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1362, in set_contents_from_filename
    encrypt_key=encrypt_key)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1293, in set_contents_from_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 750, in send_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 951, in _send_file_internal
    query_args=query_args
  File "/usr/lib/python2.7/dist-packages/boto/s3/connection.py", line 664, in make_request
    retry_handler=retry_handler
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1071, in make_request
    retry_handler=retry_handler)
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1030, in _mexe
    raise ex
socket.error: [Errno 104] Connection reset by peer
```

S3 “Requester Pays”

Normally the bucket owner pays for access fees.

With Requester Pays, the requester pays.

- No anonymous access.
- No charge to download within EC2
- No BitTorrent or SOAP

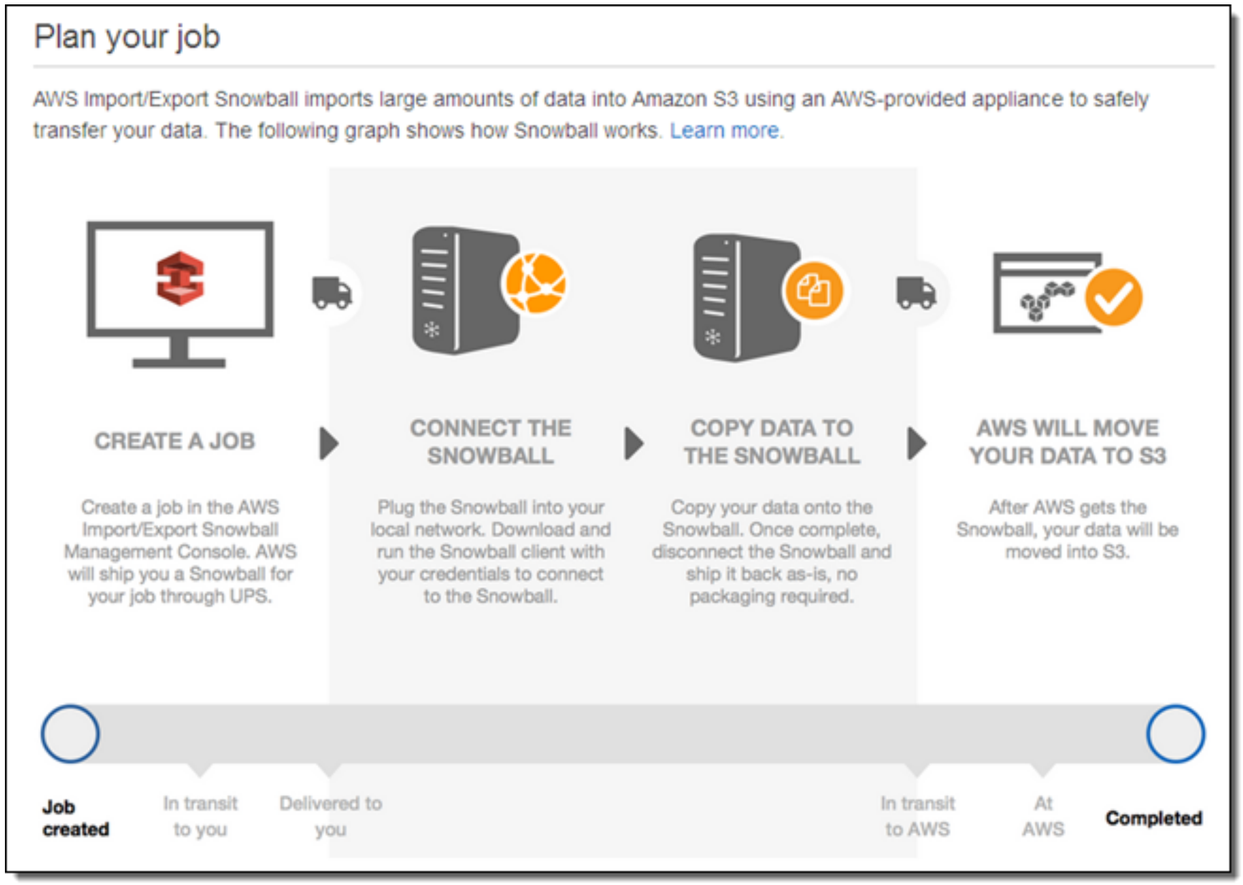
“DevPay” lets you sell your content

- Requestors are charged to access data.
- Can be combined with Requester Pays
  - https://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html
  - http://docs.aws.amazon.com/AmazonDevPay/latest/DevPayDeveloperGuide/S3RequesterPays.html

Amazon Snowball

- https://aws.amazon.com/blogs/aws/aws-importexport-snowball-transfer-1-petabyte-per-week-using-amazon-owned-storage-appliances/

Amazon Snowball



- https://aws.amazon.com/blogs/aws/aws-importexport-snowball-transfer-1-petabyte-per-week-using-amazon-owned-storage-appliances/

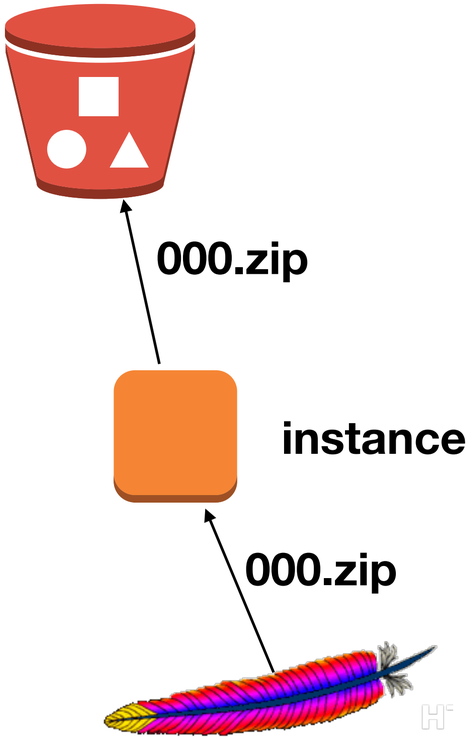
Uploading data to S3

S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - *Get the file from web server.*
  - *Send the file to S3*

How would you do this?



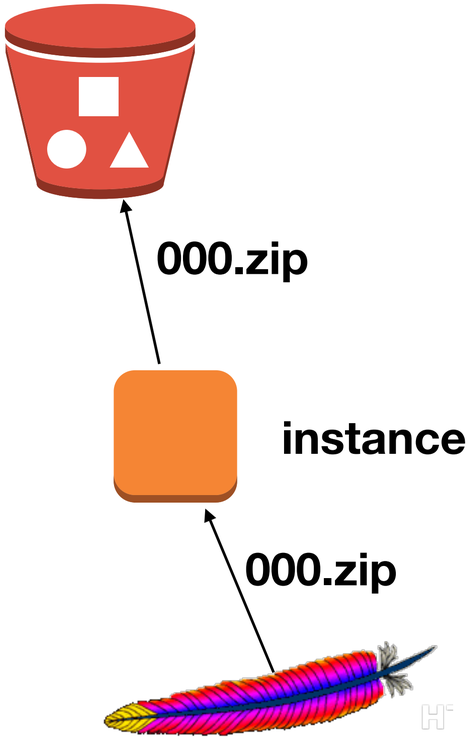
S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - *Get the file from web server.*
  - *Send the file to S3*

Potential problems:

- Download might be interrupted.
- Upload might be interrupted.
- Server might crash.



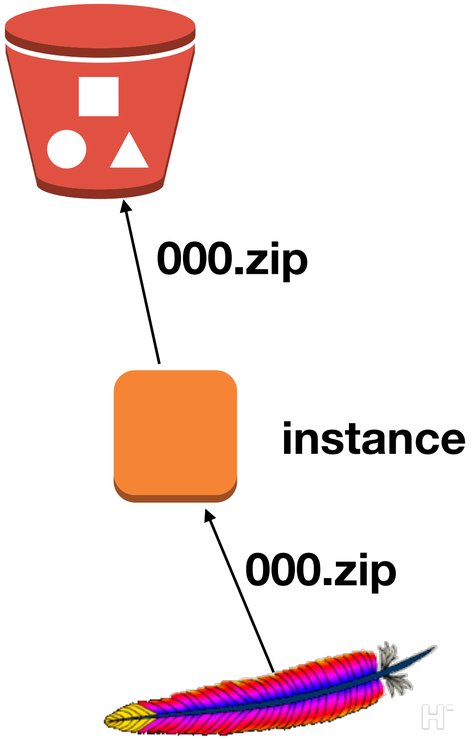
S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - —*Get the file from web server.*
  - —*Send the file to S3*

Revised approach:

- For each file N:
  - —*Get the size of file N from the WWW server*
  - —*If file N is not on S3, or if it is the wrong size:*
  - — *If file N is not on the instance, or is the wrong size:*
  - — *Download the file from the WWW server to the instance*
  - — *Upload the file to S3*



Notice that this is “indempotent”

- Tolerant to failure and restarting from beginning at any point.
- Tolerant to being run a number of times.

copy(n): Download the zipfile and upload the parts

```
def copy(n):
    fname = "%03i.zip" % n

    # get the remove file length
    url = 'http://digitalcorpora.org/corp/files/govdocs1/zipfiles/'+fname
    u = urllib2.urlopen(url)
    meta = u.info()
    file_size = int(meta.getheaders("Content-Length")[0])

    # get the key in the bucket
    key = bucket.lookup('zipfiles/'+fname)
    if key and key.exists and key.size==file_size:
        print("{} exists and is correct size ({:,}B)".format(fname,file_size))
        return (fname,0,0)
    if not key:
        key = bucket.new_key('zipfiles/'+fname)

    # Download the file if we don't have it
  if os.path.exists(fname)==False or os.path.getsize(fname)!=file_size:
        print("Downloading {}".format(url))
        block_sz = 65536
        with open(fname,"wb") as f:
            while True:
                buffer = u.read(block_sz)
                if not buffer:
                    break
                f.write(buffer)

    # Upload the file
    print("Uploading s3://{}/{}".format(bucket_name,fname))
    key.set_contents_from_filename(fname)
    key.set_acl('public-read')
    print("Uploaded {} {:,}B in {}s".format(fname,file_size,t1-t0))

    # Finally remove the uploaded file
    os.unlink(fname)
    return (fname,total_time, file_size)
```

note: total_time calculation removed

First driver program

```
if __name__=="__main__":
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('num',type=int,nargs='+')
   args = parser.parse_args()
    print(args.num)
    if len(args.num)==1:
        a = args.num[0]
        b = args.num[0]
    else:
        (a,b) = args.num[0:2]
        print(a,b)
    total_t  = 0
    total_sz = 0

    results = []
    start_time = time.time()
    for i in range(a,b+1):
        results.append(copy(i))

    end_time = time.time()
    real_time = end_time - start_time
    total_time = sum([r[1] for r in results])
    total_bytes = sum([r[2] for r in results])
    if total_time==0:
        print("nothing uploaded")
    else:
        print("Total uploaded {:,}MB in {}s, {:,}MB/sec".format(total_bytes/1E6,total_time,total_bytes/total_time/1E6))
        print("Effective upload: {:,}MB/sec in {} sec".format(total_bytes/real_time/1E6,real_time))
```

Run single-threaded on a t2.micro …
vCPU=1, CPU Credits/hour=6, Mem=1GiB, EBS-Only, Low Net

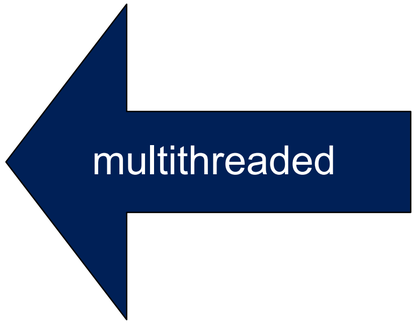
```
files/028/028753.txt uploaded
028/028754.txt
files/028/028754.txt uploaded
028/028755.txt
files/028/028755.txt uploaded
028/028756.txt
files/028/028756.txt uploaded
028/028757.txt
files/028/028757.txt uploaded
028/028758.txt
Traceback (most recent call last):
  File "govdocs.py", line 56, in <module>
    putzipparts(tfn)
  File "govdocs.py", line 37, in putzipparts
    data = z.open(zname,"r").read()
  File "/usr/lib64/python2.7/zipfile.py", line 630, in read
    data = self.read1(n)
  File "/usr/lib64/python2.7/zipfile.py", line 684, in read1
    max(n - len_readbuffer, self.MIN_READ_SIZE)
MemoryError
$
```

Total time to upload 28 files ≈ 10 minutes (until I interrupted it)

To upload 1000 files would take: 28÷10 × 1000 ≈ 2800 minutes ≈ 2 days

Modified driver program supports multithreading with multiprocessing Pool

```
if __name__=="__main__":
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('num',type=int,nargs='+')
    parser.add_argument('--multi',type=int)
    args = parser.parse_args()
    print(args.num)
    if len(args.num)==1:
        a = args.num[0]
        b = args.num[0]
    else:
        (a,b) = args.num[0:2]
        print(a,b)
    total_t  = 0
    total_sz = 0

    results = []
    start_time = time.time()
    if args.multi:
        from multiprocessing import Pool
        print("range: {} to {}".format(a,b+1))
        p = Pool(args.multi)
        results = p.map(copy,range(a,b+1))
    else:
        for i in range(a,b+1):
            results.append(copy(i))

    end_time = time.time()
    real_time = end_time - start_time
    total_time = sum([r[1] for r in results])
    total_bytes = sum([r[2] for r in results])
    if total_time==0:
        print("nothing uploaded")
    else:
        print("Total uploaded {:,}MB in {}s, {:,}MB/sec".format(total_bytes/1E6,total_time,total_bytes/total_time/1E6))
        print("Effective upload: {:,}MB/sec in {} sec".format(total_bytes/real_time/1E6,real_time))
```



Run with 1-6 threads to upload ZIP files in 50-file batches

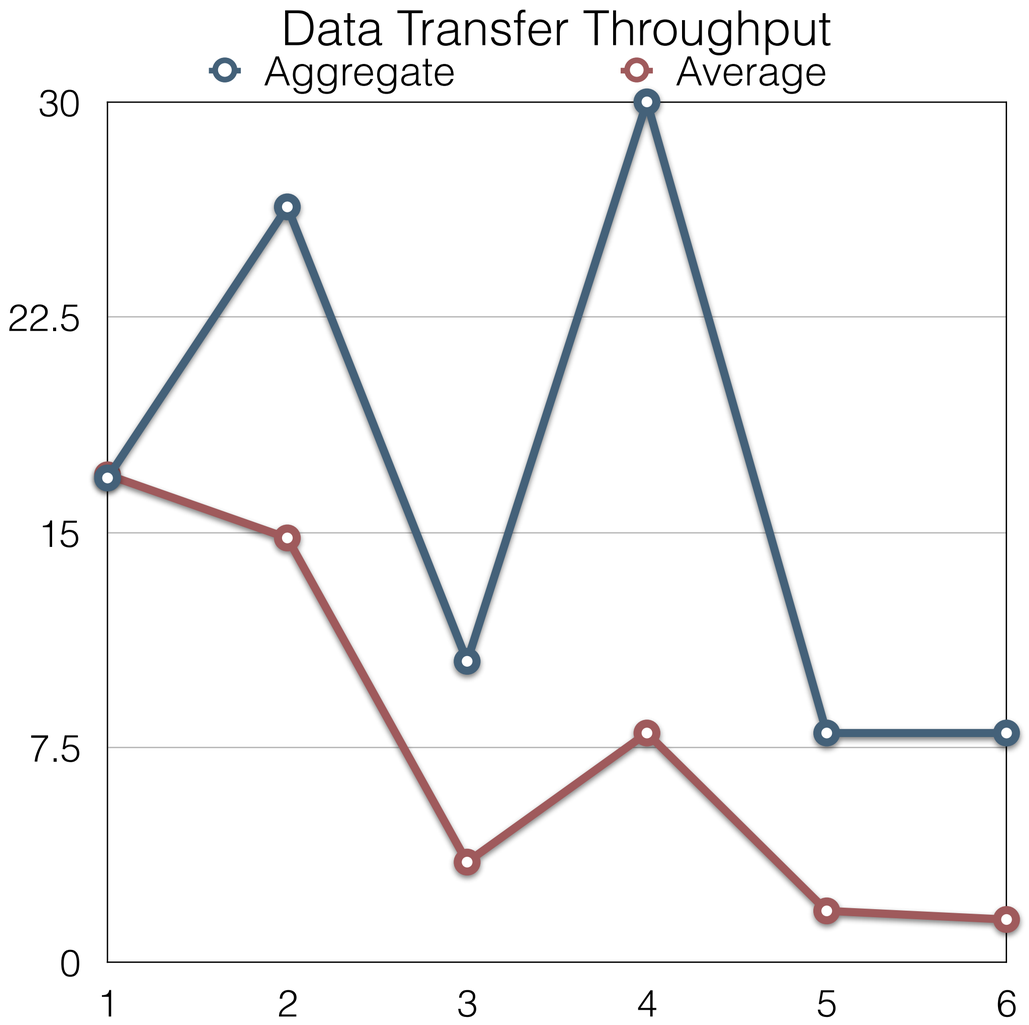
Verified multi threading:

simsong — ec2-user@ip-172-30-1-99:~ — ssh ec2-user@52.91.31.86 — 84×28

```
top - 23:26:35 up 23 min,  3 users,  load average: 1.62, 1.09, 0.50
Tasks: 301 total,   2 running, 299 sleeping,   0 stopped,   0 zombie
Cpu(s):  2.7%us,  0.1%sy,  0.0%ni, 97.0%id,  0.2%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:  165055516k total,  7487684k used, 157567832k free,    16776k buffers
Swap:        0k total,        0k used,        0k free,  6305548k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4621 ec2-user  20   0  222m  22m 5752 S 26.3  0.0   0:51.90 python27
 4624 ec2-user  20   0  240m  41m 5756 S 18.3  0.0   0:16.08 python27
 4146 ec2-user  20   0  163m  20m 7740 S 12.3  0.0   0:22.55 emacs
 4630 ec2-user  20   0  236m  36m 5756 S 10.0  0.0   0:17.23 python27
 4622 ec2-user  20   0  271m  66m 5728 R  9.0  0.0   0:38.47 python27
 4626 ec2-user  20   0  269m  72m 5756 S  8.3  0.0   0:27.37 python27
 4625 ec2-user  20   0  255m  46m 5756 S  8.0  0.0   0:15.27 python27
 4629 ec2-user  20   0  232m  31m 5748 S  8.0  0.0   0:18.10 python27
 4627 ec2-user  20   0  243m  34m 5756 S  7.0  0.0   0:15.05 python27
 4628 ec2-user  20   0  238m  34m 5756 S  6.6  0.0   0:15.26 python27
 4623 ec2-user  20   0  237m  38m 5728 S  5.7  0.0   0:19.45 python27
 4094 ec2-user  20   0  122m 2896 2172 S  0.3  0.0   0:00.70 screen
 4715 ec2-user  20   0 15400 2336 1872 R  0.3  0.0   0:00.04 top
    1 root      20   0 19620 2596 2264 S  0.0  0.0   0:01.58 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u256:0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u257:0
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.60 rcu_sched
```

Performance improved from 1 to 2 threads, then decreased. (contention between threads.)

		Per Thread		Agregate	
Threads	Data Uploaded	Total Clock Time Σ(threads)	Avg throughput per thread	Wall Clock Time	Aggregate Througput
1	18GB	1087	17.0 MB/s	1087	16.89 MB/s
2	16 GB	1288	14.8 MB/s	644	26.34 MB/s
3	18 GB	1725	3.5 MB/s	575	10.5 MB/s
4	17 GB	2244	8 MB/s	561	30 MB/s
5	17 GB	9364	1.8 MB/s	1951	8 MB/s
6	17 GB	10,741	1.5 MB/s	1999	8 MB/s



Many core files.
Some of the sub processes were crashing—lack of memory!

```
$ ls -l
total 1127728
drwx------  4 ec2-user ec2-user      4096 Dec  2 02:00 anly502/
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:33 core.27872
-rw-------  1 ec2-user ec2-user   1404928 Dec  5 04:33 core.27880
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:34 core.27942
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:34 core.27998
-rw-------  1 ec2-user ec2-user   1400832 Dec  5 04:38 core.28025
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:39 core.28078
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:49 core.28138
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:49 core.28193
-rw-------  1 ec2-user ec2-user   1400832 Dec  5 04:49 core.28208
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:50 core.28266
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:50 core.28326
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:51 core.28381
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:52 core.28515
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:53 core.28607
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:53 core.28662
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:55 core.28715
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:55 core.28771
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:55 core.28824
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:58 core.28877
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:59 core.28932
-rw-------  1 ec2-user ec2-user    573440 Dec  5 05:00 core.28989
-rw-------  1 ec2-user ec2-user    573440 Dec  5 13:31 core.29694
-rw-------  1 ec2-user ec2-user    573440 Dec  5 16:05 core.30033
-rw-------  1 ec2-user ec2-user   1400832 Dec  5 18:04 core.30219
-rw-------  1 ec2-user ec2-user   1400832 Dec  5 18:04 core.30223
-rw-------  1 ec2-user ec2-user    573440 Dec  5 18:26 core.30427
-rw-------  1 ec2-user ec2-user    573440 Dec  2 03:18 core.3437
-rw-------  1 ec2-user ec2-user    573440 Dec  2 23:59 core.5249
-rw-------  1 ec2-user ec2-user    573440 Dec  3 00:04 core.5331
-rw-------  1 ec2-user ec2-user    573440 Dec  4 03:22 core.8704
-rw-------  1 ec2-user ec2-user    573440 Dec  5 04:01 core.936
```

Spin up a high capacity machine.

Step 2: Choose an Instance Type

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☐	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate
☐	General purpose	m4.xlarge	4	16	EBS only	Yes	High
☐	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
☐	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
☑	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit

Purchasing option ☑ Request Spot instances

Current price

us-east-1a	5.000
us-east-1b	0.4404
us-east-1c	0.400
us-east-1e	0.4205

Maximum price $ 0.50

Normally $2.52/Hour

Spot price: $0.50/Hour!

Spin up a high capacity machine.

Step 2: Choose an Instance Type

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☐	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate
☐	General purpose	m4.xlarge	4	16	EBS only	Yes	High
☐	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
☐	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
☑	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit

Purchasing option ☑ Request Spot instances

Current price

us-east-1a	5.000
us-east-1b	0.4404
us-east-1c	0.400
us-east-1e	0.4205

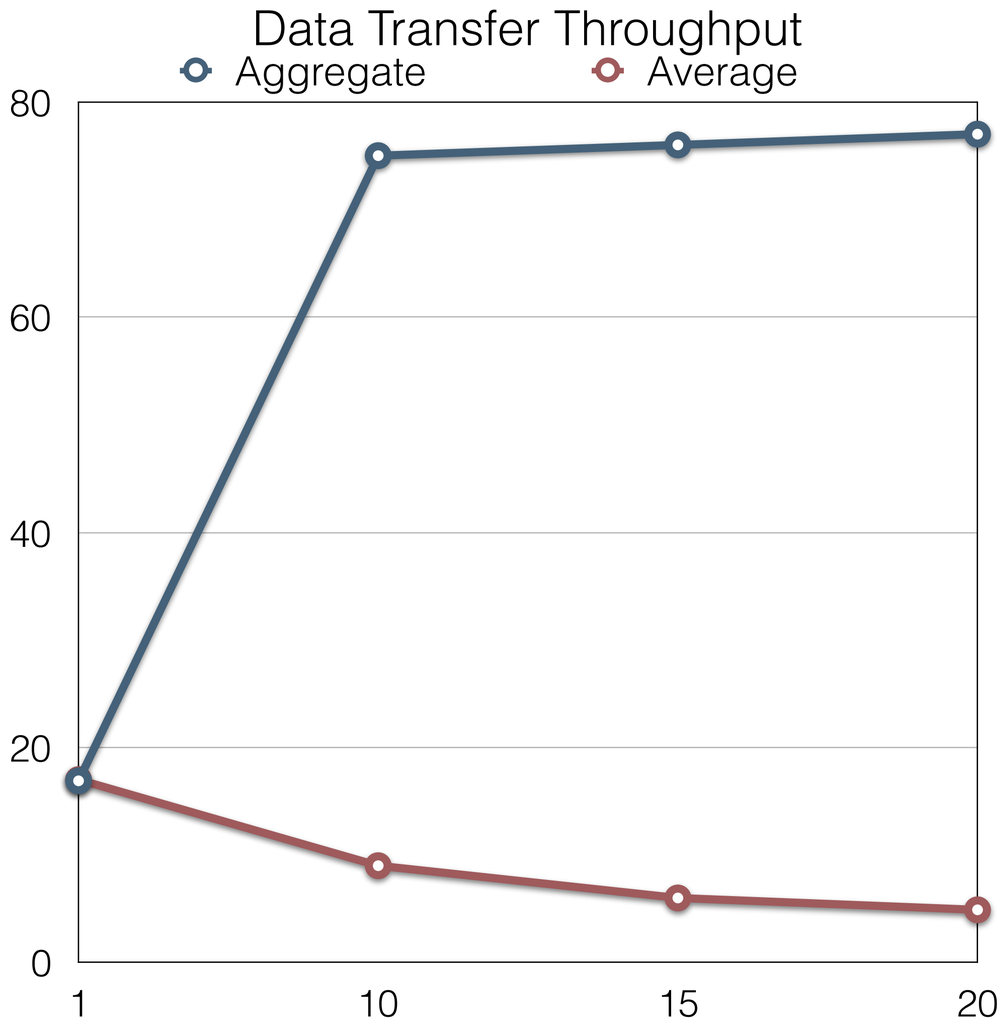
Maximum price $ 0.50

Normally $2.52/Hour

Spot price: $0.50/Hour!

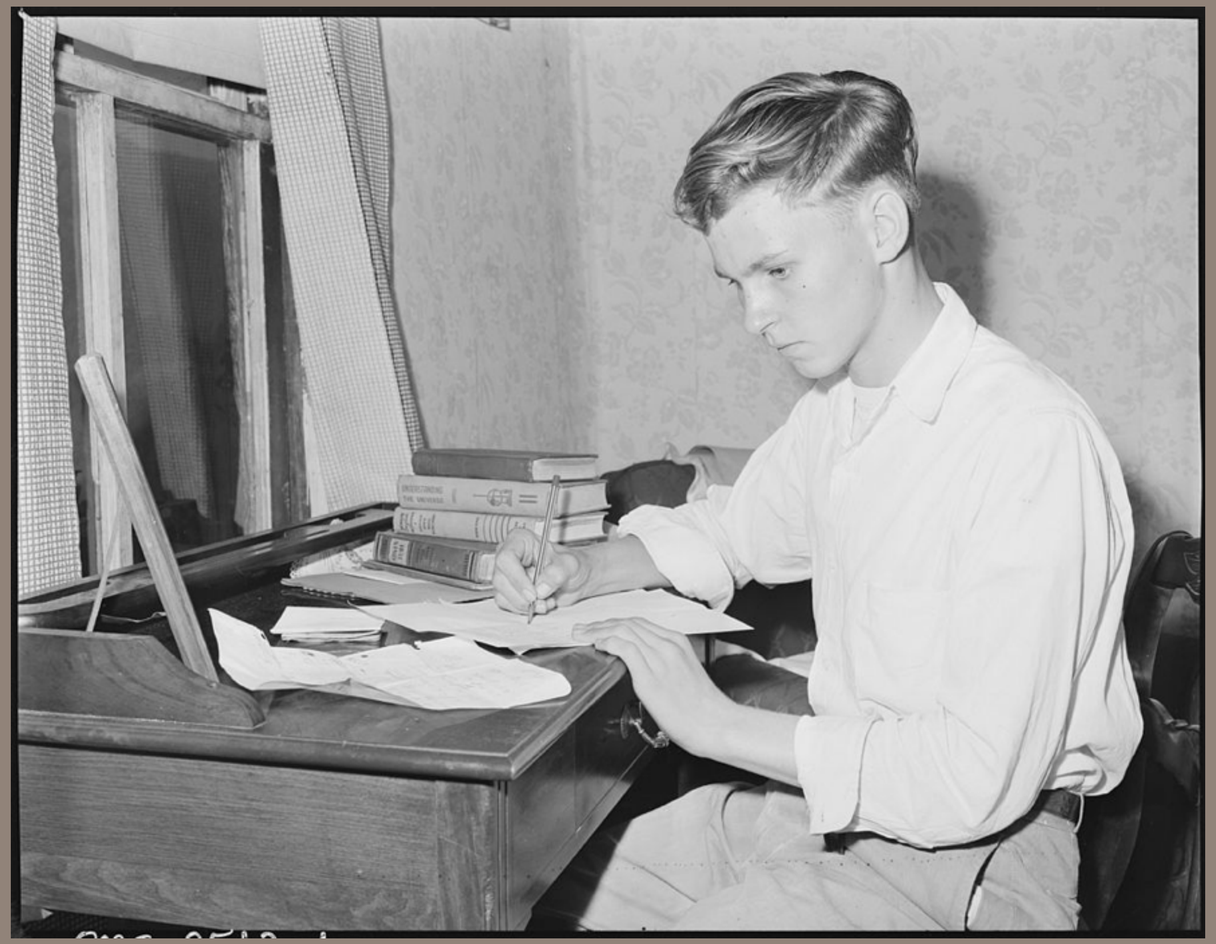
Performance on the faster machine: much better

		Per Thread		Agregate	
Threads	Data Uploaded	Total Clock Time Σ(threads)	Avg throughput per thread	Wall Clock Time	Aggregate Througput
1	18GB	1087	17.0 MB/s	1087	16.89 MB/s
2	16 GB	1288	14.8 MB/s	644	26.34 MB/s
10	15 GB	1623	9 MB/s	194	75 MB/s
15	14 GB	2219	6 MB/s	186	76 MB/s
20	17 GB	3446	4.9 MB/s	224	77 MB/s



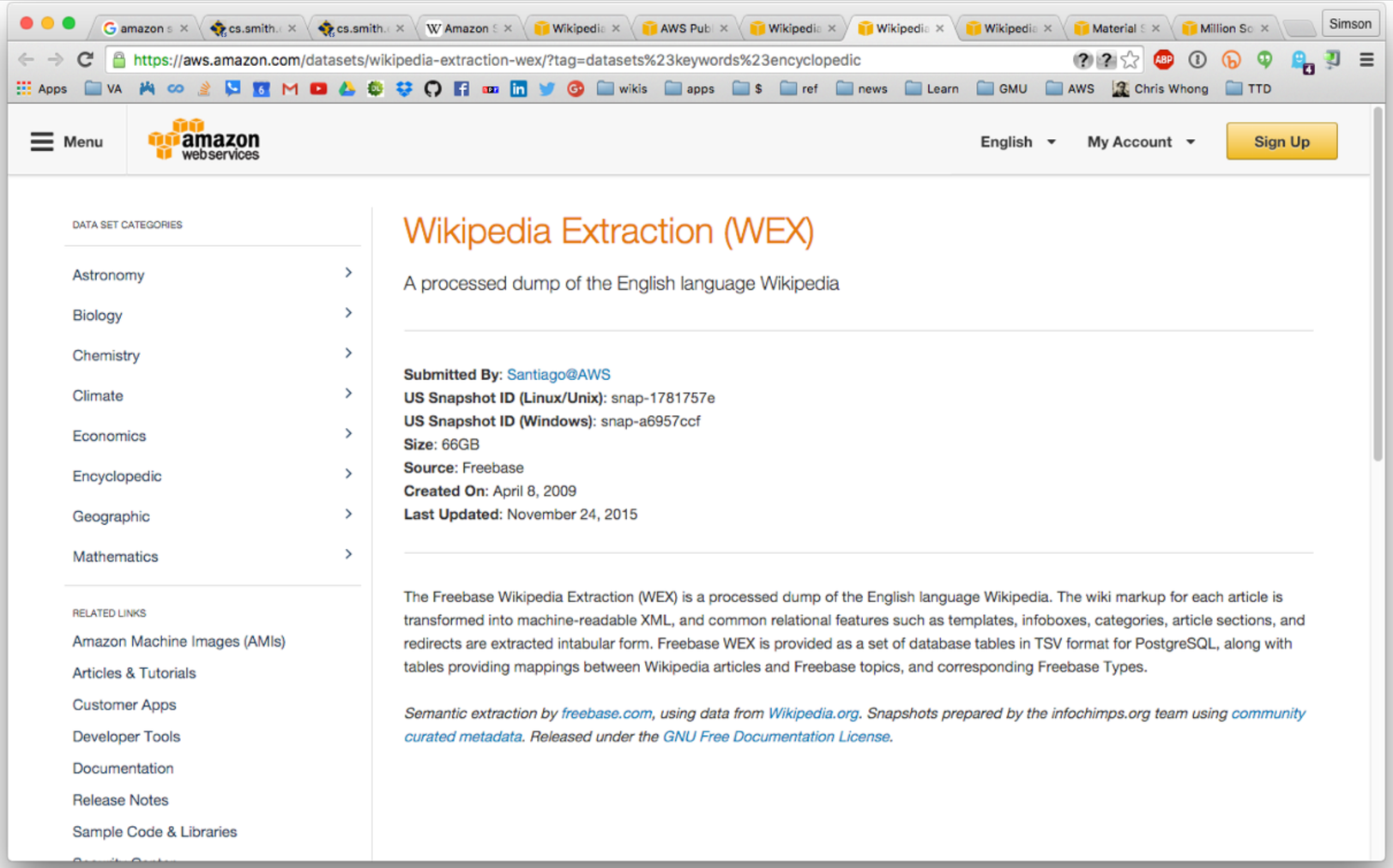
Notice: adding more threads improved performance, but not beyond 75 MB/sec

- New bottleneck: remote server?



Homework

For PS03, we're doing something big



http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-public-data-sets.html

To use a public snapshot

First you need to create a private EBS volume from the public snapshot:

Submitted By: Santiago@AWS
US Snapshot ID (Linux/Unix): snap-1781757e
US Snapshot ID (Windows): snap-a6957ccf
Size: 66GB
Source: Freebase
Created On: April 8, 2009
Last Updated: November 24, 2015

```
$ ec2-create-volume --snapshot snap-1781757e --availability-zone=us-east-1b
VOLUME          vol-44d495a7          66          snap-1781757e         us-east-1b creating   2015-12-07T03:57:43+0000
standard                    Not Encrypted
```

Other snapshots.

Wikipedia XML Data — 500GB

- https://aws.amazon.com/datasets/wikipedia-xml-data/?tag=datasets%23keywords%23encyclopedic

Wikipedia Extraction (WEX) — 66GB of Wikipedia

- https://aws.amazon.com/datasets/wikipedia-extraction-wex/?tag=datasets%23keywords%23encyclopedic

Wikipedia Page Traffic Statistics — 320GB

- https://aws.amazon.com/items/2596?externalID=2596

Wikipedia Page Traffic Statistic V3 — 150 GB of sample data

- https://aws.amazon.com/datasets/wikipedia-page-traffic-statistic-v3/

Material Safety Data Sheets — 230K MDSs, 3GB

- https://aws.amazon.com/datasets/material-safety-data-sheets/

Million Song Dataset — 500GB of songs and features

- https://aws.amazon.com/datasets/million-song-dataset/

Additional Resources

Resources for understanding EC2

Current instances:

- http://www.ec2instances.info/

Getting started with AWS and Python:

- https://aws.amazon.com/articles/Python/3998

mrjob:

- Donald Miner PyCon 2015 - https://www.youtube.com/watch?v=b8HLYUp_fA8

AWS Slideshow about EMR:

- http://www.slideshare.net/AmazonWebServices/deep-dive-amazon-elastic-map-reduce

S3 and Hadoop performance:

- http://blog.mortardata.com/post/58920122308/s3-hadoop-performance
- https://aws.amazon.com/blogs/aws/amazon-s3-performance-tips-tricks-seattle-hiring-event/

Nice explanation of Hadoop joins:

- https://chamibuddhika.wordpress.com/2012/02/26/joins-with-map-reduce/
- http://blog.matthewrathbone.com/2013/02/09/real-world-hadoop-implementing-a-left-outer-join-in-hadoop-map-reduce.html

AWS re:Invent (Amazon's trade show)

https://reinvent.awsevents.com/

https://www.youtube.com/user/AmazonWebServices/Cloud

"State of the Union: AWS Storage Services" — http://bit.ly/1khmDP6

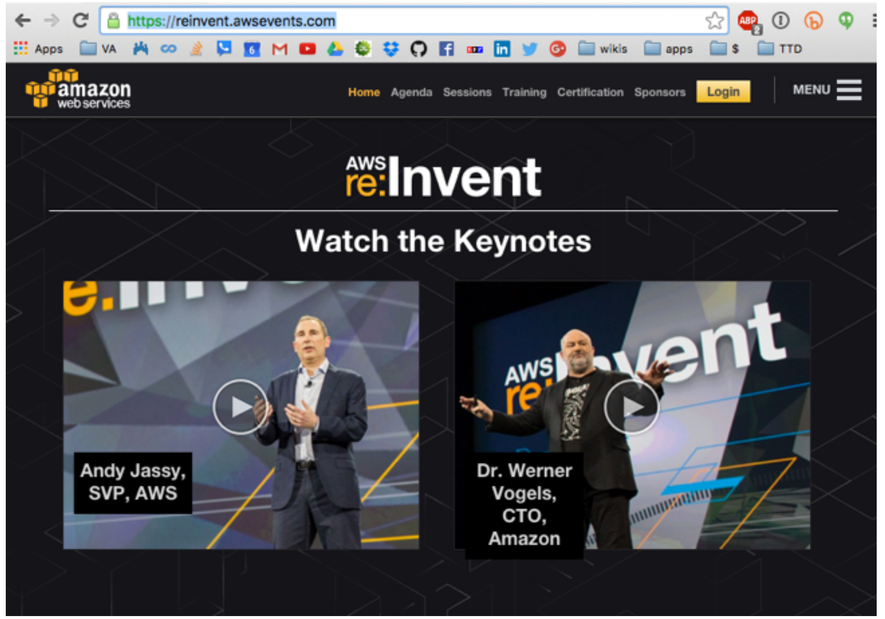
"Self-service Cloud Services" — http://bit.ly/1khmlm5

"Real-World Smart Applications with Amazon Machine Learning"

- http://bit.ly/1khmLyj

"Amazon RDS for PostgreSQL"

- http://bit.ly/1khmM5g



git - Get to know it

Use git for:

- Storing your work
- (soon: Submitting your homework programs)

Git tutorials:

- https://try.github.io/levels/1/challenges/1
- http://git-scm.com/docs/gittutorial
- https://www.atlassian.com/git/tutorials/

Git GUI: Atlassan SourceTree:

- https://www.sourcetreeapp.com/